Developing usability method for agile software development – Case study on RISE for Traffica

Ari Koivuniemi

University of Tampere
School of Information Sciences
Interactive Technology
M.Sc. Thesis
Supervisor: Markku Turunen
April 2013

University of Tampere
School of information sciences
Interactive Technology
Ari Koivuniemi: Developing usability method for agile software development – Case study on RISE for Traffica
M.Sc. Thesis, 55 pages, 4 appendix pages
April 2013

This study presents the development work done for RISE for Traffica, a tool for managing network element adaptations that are used by Nokia Siemens Networks' real-time network monitoring system Traffica. The goal of this study was to develop the new tool and come up with an inexpensive and fast usability method that would verify the product's usability and provide feedback for its further development.

RISE for Traffica is an expansion to existing data management system RISE allowing it to be used with Traffica. RISE for Traffica is being developed to replace several currently used tools and change the adaptation creation process with this single system. The change meant that potential problems with user acceptance were foreseen. Usability was used as the approach towards the goal of making RISE for Traffica an accepted replacement for the current solutions.

Five usability tests were conducted during the work for the first release version of RISE for Traffica. The first test was summative in nature and its results provided the first glimpse on the possible future feedback. The test method was developed towards more explorative in each subsequent test. A satisfactory level was reached in the fifth usability test. The advances in developing the testing method were measured by the quantity and quality of resulting findings and feedback.

The tests resulted in over two hundred findings and improvement targets. The findings were distributed equally between usability problems and content related issues, so the goal set for usability and product development was reached. Using the same meter it can be concluded that the development of the usability method was successful as each test provided answers to the questions that were under work at that time.

Key words and terms: usability, usability testing, product development through usability.

Tampereen yliopisto
Informaatiotieteiden yksikkö
Vuorovaikutteinen teknologia
Ari Koivuniemi: Developing usability method for agile software development – Case study on RISE for Traffica
Pro gradu – tutkielma, 55 sivua, 4 liitesivua
Huhtikuu 2013

Tässä työssä käsitellään verkkoelementtiadaptaatioiden hallintaan tarkoitetun RISE for Traffica – työkalun kehitystyötä. Työ tehtiin Nokia Siemens Networksin organisaatiossa, jossa kehitetään reaaliaikaiseen verkon monitorointiin tarkoitettua Traffica -järjestelmää, jonka käyttöön adaptaatiot tulevat. Työn tarkoituksena oli kehittää uutta työkalua ja samalla myös edullinen ja nopea menetelmä, jolla voidaan verifioida tuotteen käytettävyys ja saada palautetta sen jatkokehitykseen.

RISE for Traffica on yrityksen käytössä olevaan RISE -datanhallintajärjestelmään tuleva laajennus, joka mahdollistaa sen käytön Traffican kanssa. Uusi tapa hallinnoida adaptaatiodataa tuo mukanaan myös täysin uuden työskentelytavan, minkä vuoksi RISE for Traffican kehitystyössä oli ensisijaisen tärkeää varmistaa tuotteen hyväksyntä loppukäyttäjien keskuudessa. Tätä lähdettiin tavoittelemaan käytettävyyden kautta.

Projektin käynnistymisen ja RISE for Traffican ensimmäisen julkaisun välillä sen eri kehitysversioille tehtiin viisi käytettävyystestiä. Ensimmäinen testi oli luonteeltaan tuotetta arvioiva. Sen tulokset antoivat suuntaa, minkälaista palautetta oli mahdollista saada. Testiä ei kuitenkaan pidetty vielä riittävänä, joten testausmenetelmiä lähdettiin kehittämään kohti viidennessä testissä tavoitettua kokeellisempaa testimuotoa. Testimenetelmien kehityksen mittarina pidettiin saavutettujen löydösten ja palautteen määrää ja laatua.

Työn tuloksina testeissä löydettiin yli kaksisataa ongelmaa ja kohdetta, joita voitiin parantaa RISE for Traffican seuraaviin iteraatioihin. Löydökset jakaantuivat hyvin tasaisesti käytettävyysongelmien ja tuotteen sisältösidonnaisten ongelmakohtien välillä, joten testeille asetetut tavoitteet käytettävyyden ja tuotekehityksen alalta saavutettiin. Käytettävyysmenetelmän kehityksessä voidaan samalla mittarilla katsoa myös onnistutun tulosten vastatessa kehitysvaiheessa esitettyihin kysymyksiin.

Avainsanat ja -sanonnat: käytettävyys, käytettävyystestaus, tuotekehitys käytettävyyden kautta.

Acknowledgements

This work has provided me with more surprises than I ever could think of in its beginning. While the process has surely taken its time: at first to get the topic defined and the actual work underway took months, and even after that the writing process took almost a year with all the pauses and motivation breakdowns. Still this time has been educational, not only professionally but also in personal growth as well. I suppose putting it all in one word the best choice would be adventure.

Since I was never alone on this adventure I would like to express my deepest gratitude to some amazing people without whom this work would not be.

First of all I want to thank Ossi Ahola who for some reason thought I might be the right man for this job. Since then it must not have been easy to always have faith in the project and its eventual outcome, but fortunately Ossi has great amounts of patience and faith.

Thank you to my supervisor Markku Turunen. This probably was not the most typical thesis work to supervise or to follow in deed, but still I managed to get good advice whenever I needed one.

No words of thanks could ever describe or pay back what I got from the former XD-team. Jenni, Jesse and Timo, we had our ups and downs but you guys made me a better person. I am forever indebted, thank you.

Special thanks to Mikko Pihlajamäki for mentoring me in literally everything and to Lauri Pietarinen for providing the no doubt necessary kick to finish this thesis. Big thanks go to everyone in Traffica R&D for always managing to spare some time and interest in my endeavors to try and understand.

Friends, what can I say, without you I would not be here. I'm yours.

And finally to my family. I know it has been a long and winding road to follow but maybe it is finally getting somewhere. Thanks for believing.

Ari Koivuniemi
Tampere, January 31st 2013.

Contents

Glossary and Abbreviations

Adaptation	Metadata that is used to configure the software to handle new network elements or service types
CuDo	Customer Documentation
Developer	Professional involved with the creation of the system
GDD	Goal-Directed Design
GUI	Graphical User Interface
HCI	Human Computer Interaction
NE	Network Element
Network Monitoring System	A tool, or a box of tools, used for monitoring the network
NOLS	Nokia Siemens Networks Online Services, wide online service concept for infrastructure business including operators, enterprises, and value added resellers
NSN	Nokia Siemens Networks
RISE	Reference Information Service Environment: tool for creating, managing, editing, and publishing metadata and documentation for alarms and counters
RISE for Traffica or R4T	A new section to RISE designed to be used in Traffica development work
RITE	Rapid, Iterative Testing and Evaluation: a discount usability method
RTT	Real-time Traffic: a report type for Traffica
SVN	Subversion, a software versioning and revision control system
Traffica	A real-time network monitoring system developed by Nokia Siemens Networks
User	Person who will actually work with the system or product being built
UX	User Experience
XP	Extreme Programming

1. Introduction

People today are accustomed to ubiquitous technology: there are technical devices big and small all around us almost constantly. Technological breakthroughs have become so common that words high tech have virtually lost their meaning as people no longer wonder at the technology itself but are more interested in what they can do with it and – especially since the advent of social media - with whom. One of the latest questions has also been how: everyone likes to choose for themselves how they do things from paying at the local market to communicating with friends. Many people now want to have an experience from using technology or software.

Still, selecting the best tools for the job at hand is more often than not based on other factors. Many professionals require e.g. software that performs its specific task rather with efficiency than with subjective pleasantness. As Cooper stated [2007] for the professionals the product design starts with its purpose. The user experience (UX) design starts with good usability, making the product as easy and as efficient to use as possible.

This paper discusses a project to develop a software system with good usability for professional use. The work presented here is a part of on-going development project at Nokia Siemens Networks (NSN). The project described here has reached its first release phase and the results reflect the development process in an agile environment so far.

The product whose development process is followed in the study is RISE for Traffica (R4T). RISE itself is a system intended for creating, managing, editing, and publishing various kinds of metadata and documentation. Traffica is NSN's real-time network monitoring system that uses its own specific data. The project is about specifying and implementing the required changes to RISE system in order to make it suitable for storing Traffica-related data.

The motivation for this study was to discover the best method for developing a software product of this sort through methods of usability engineering. The role of usability in the project is to make R4T as easy and efficient a tool as possible and to achieve acceptance from its end users.

A number of different usability methods and approaches are selected as the starting point. They are employed to use and the achieved results are reported. Because RISE for Traffica will not be ready for release during this study it cannot be fully tested for acceptance. Therefore user acceptance is measured indirectly through interviews. After reporting the test results this paper analyzes the used methods for their efficiency and how well they progress this kind of project. The results reflect this progress. The goal of the work is to come up with method that best suits the needs of agile software development environment of NSN's Traffica program. And hopefully increase the awareness of usability within the program in the process.

Presented here is a case study that focuses first on discovering the best usability methods that might be employed in the project of this nature, then implementing these methods and sorting out their results. In the discussion part achieved results will be analyzed and the methods will be

evaluated according to how well they suited their purpose in this specific type of development process.

The structure of this paper is as follows. The first chapter introduces the motivation and background for this work. The second chapter introduces the development environment, in which this work takes place. The complex nature of the system environment is necessary to understand in order to comprehend the specific challenges for usability. These are discussed in more detail when the term usability is introduced in the third chapter. Chapter 4 introduces and discusses some usability methods that were selected for this work. In the fifth chapter the conducted tests are introduced and their results presented. Also the basis for the developed usability testing method is discussed. Following the test results they are discussed in chapter 6 along with more analysis on method development. In chapter 7 the final conclusions are presented and the overall success of the work contemplated.

2. Development environment

The project discussed in this thesis concerns the basis for the adaptation creation process for NSN's network monitoring system Traffica. The on-going project's goal is to develop a new system into Reference Information Service Environment (RISE) that would better support specifically Traffica related tasks. These tasks have previously been conducted on a system designed to be used with another product, NetAct. This process currently takes place in a number of separate phases and requires a lot of manual work and the use of different tools.

This situation has lead to compromises and temporary solutions thus eventually creating a situation of technical debt, which Cunningham [1992] introduced as a concept in his report for OOPSLA '92. Technical debt had originally to do with programming, but has since extended to include all the software engineering. Cunningham later defined technical debt as: "Things that you choose not to do now, but which will impede future development if left undone" [Cunningham, 2011].

Current workflow has been deemed as too time consuming and ineffective way of working, which inspired current project to streamline the work process. The starting point was to develop better tools for the work, which inevitably lead to different ways of working. Therefore the focus in this work should be held on the whole work process even if it sometimes means settling for less than the best single technical solution.

While this work focuses on making Reference Information Service Environment (RISE) available for Traffica use, two other essential elements of the adaptation creation process need to be explained in order to understand the concept of network adaptations and the work's context. In this chapter the workflow of creating and using network adaptations is presented through Traffica, Adaptation Toolkit and RISE.

Figure 1 explains the intended result of the renewed workflow of adaptation creation process. Currently configuration files are stored in SVN and manually fetched and edited with each new release. A version of Adaptation Toolkit, which is used to edit configuration files, exists, but is mostly obsolete. A new version of Toolkit is being developed as a part the same project to work in conjunction with RISE. The new idea is to store all the data needed (Network element interface specifications and Traffica related information) to create adaptation/configuration files in one place, RISE for Traffica. This data will then be exported as XML-files to the new Adaptation Toolkit, which in turn creates configuration files from the data.

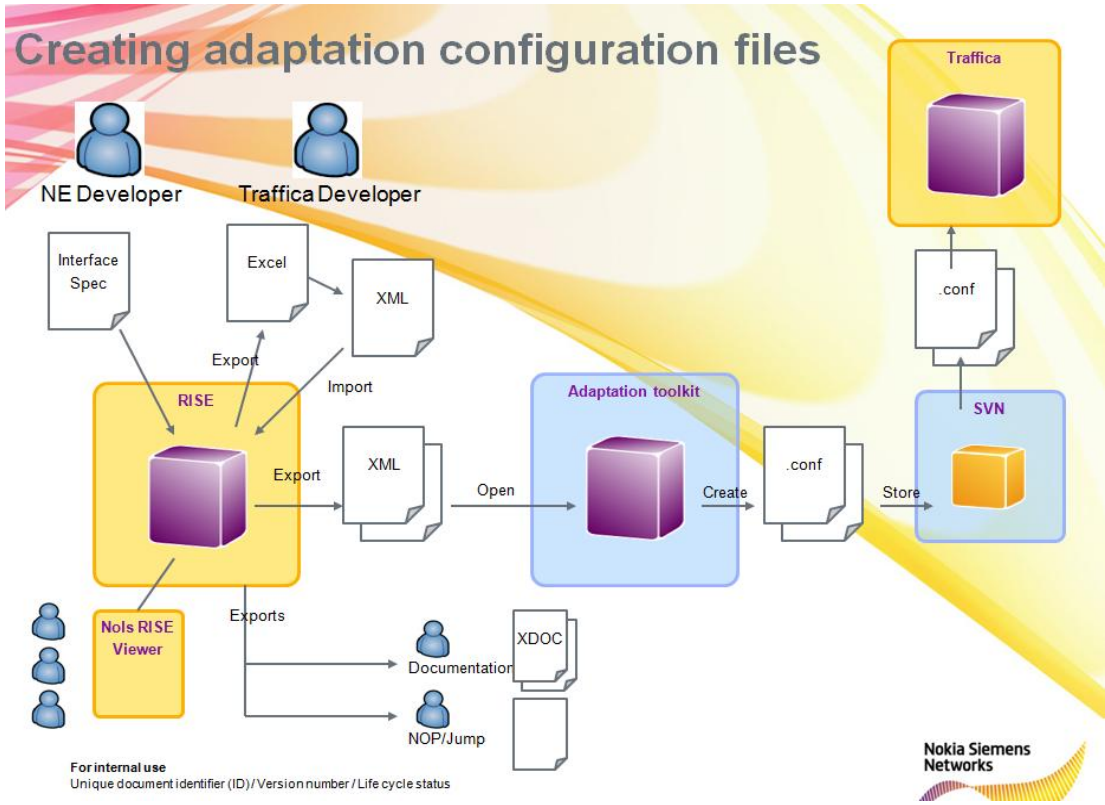


Figure 1. The goal for the new workflow of adaptation creation process.

2.1. Network Elements and Adaptations

Network elements (NE) are the building blocks of telecommunication networks. A network element is by definition: “System that can be managed, monitored, or controlled in a telecommunications network that has one or more standard interfaces, and is identified by a unique management address” [NE, Nokia term bank]. In the scope of this work, network elements can be simplified to be elements that provide data of the traffic in the selected networks.

There are a number of different network elements and each of them has their own interface. This means that a network monitoring system like Traffica needs to be configured to accept and handle data in various forms. This process is called adapting the data and hence the Traffica configurations for different NE’s are called network element adaptations.

Adaptations are metadata and technically, this metadata is configuration data, not software. Its purpose is to configure software, in this case Traffica, to handle new network elements or service types. Adaptation metadata can be for example in the format of an XML file. Adaptation metadata can be used to configure for example applications, user interfaces, databases, mediation components, and business rules. Both NSN and its customers can create adaptation metadata.

2.2. Traffica

Traffica (figure 2) is a real-time network monitoring system developed by Nokia Siemens Networks. It is the product that all the work described in this thesis eventually leads to.

Traffica is in short tool for monitoring real-time live traffic and subscriber activity in mobile broadband networks. Traffica provides real-time visibility over the end-users activity and service usage in the whole network or down to cell level. Traffica operator has access to visualizations for example of how much subscribers are using the services, at what time, from where in the network and what problems they might have. Other information that can be obtained through Traffica include details of activities for each subscriber, such as error codes, the usage and problems per mobile type and the activity of user groups such as roamers, home subscribers or corporate customers. This information can be used in troubleshooting for example locating and identifying problem that causes calls to drop, or finding the actual busy hours and reallocating resources.

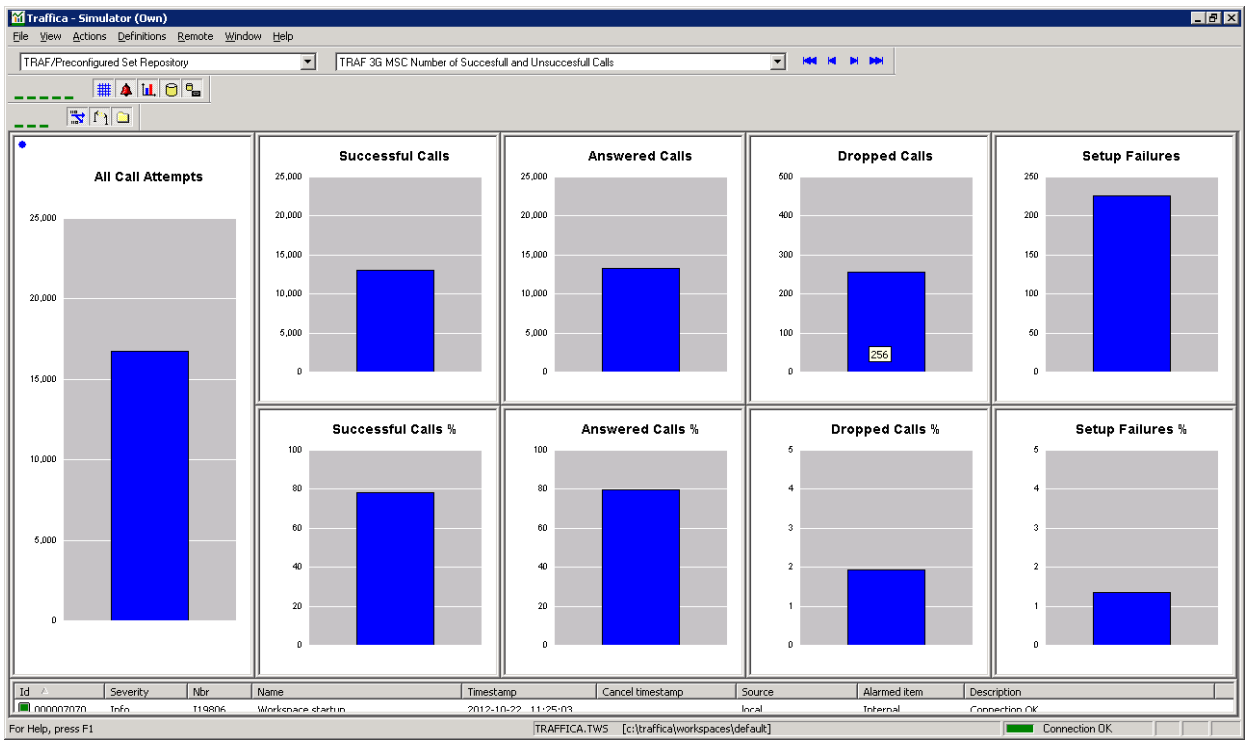


Figure 2. Different performance indicators visualized in Traffica.

Traffica collects data and information from several components, network elements that are visualized to the user. Traffica is in use all around the world and it is modified to suit each customer's needs. Different network elements provide different services, and these modifications mean that customers get certain service packages to operate Traffica.

Since Traffica is used to collect data from several different kinds of NE's it needs to be able to accept just as many formats of input data. The key issue is adapting all this different data and visualizing it correctly. This is where the adaptations for each NE come along to tell Traffica what is this data and how to interpret it.

2.3. Adaptation Toolkit

The Adaptation Toolkit is a tool that is planned to accept RISE exports as inputs and create adaptation specific configuration files to be used by Traffica. The current standalone desktop version of the Toolkit (figure 3) is complex and has become largely obsolete. A new web-based version is under development. The renewing of Adaptation Toolkit was the original starting point for this whole process, as it was considered that in order to optimize the tool's working, its input need to be optimized too. Therefore a new Traffica specific section to RISE was deemed necessary.

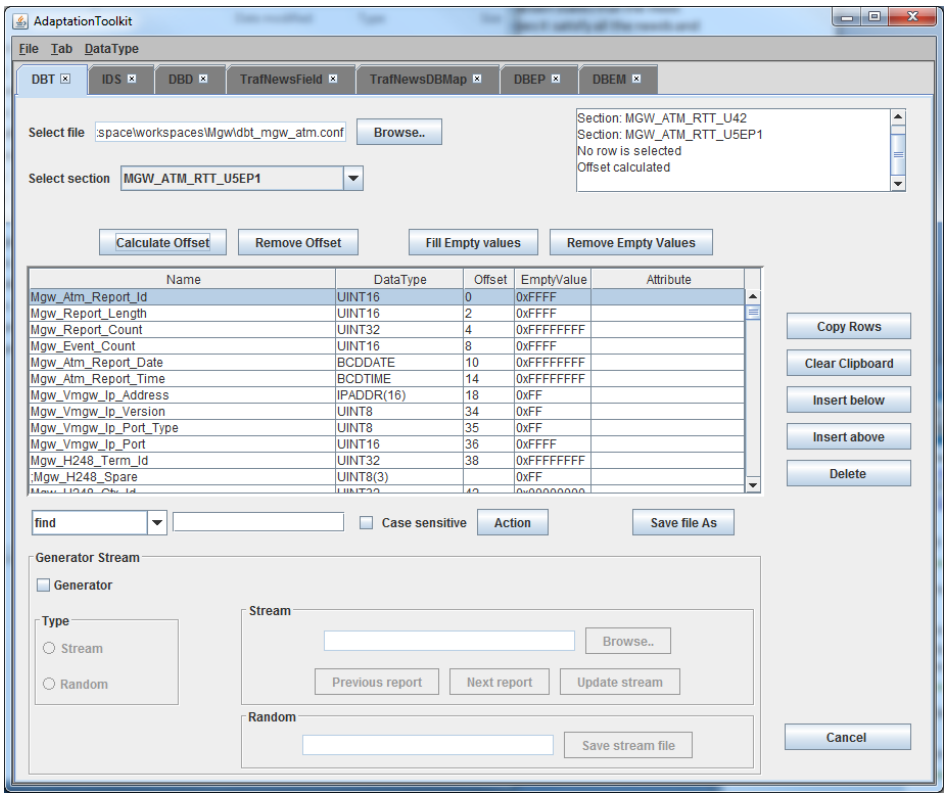


Figure 3. The old version of Adaptation Toolkit.

Adaptation Toolkit's sole purpose is to create adaptations, configuration files, to be used by Traffica. In order to create these configuration files, Toolkit needs the specifications for each different network element. All the adaptation creation data is currently gathered and processed manually, but with this project's results the data will be exported from single place, RISE, and processed into configuration files in Adaptation Toolkit.

The idea is to have RISE produce material in such shape and form that Adaptation Toolkit can use it directly to generate configuration files for Traffica. The renewing of Adaptation Toolkit was behind the need to renew RISE for Traffica-related issues. It can in fact only be implemented and properly tested once RISE is able to produce the required data.

The original vision for renewing Adaptation Toolkit was to just transform current version to work on a server and make users operate it through their web browsers. This would have been mostly the same program with possibly just some fine-polished features. During this project the

concept of Adaptation Toolkit has evolved from that to being server software. The release version will most likely be close to current ideas and included functionalities will be plain. Current work on Adaptation Toolkit focuses on its compatibility with RISE.

Figure 4 shows how much Adaptation Toolkit's GUI can be simplified with the help of input exported from RISE: practically all of the modifications and setups that were previously occupying Toolkit's UI have been transferred to RISE. This is still work in progress version and features will be added but at the moment most of the previous work steps have been implemented in RISE or have been automated and hidden from the users. It could be argued that the new Adaptation Toolkit could have been incorporated into RISE as well but for internal reasons it was still wanted as a separate, independent solution.

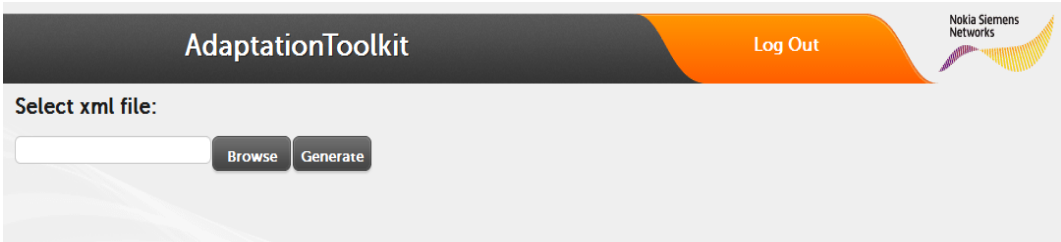


Figure 4. Current version of the new Adaptation Toolkit (January 2013).

The future plans for Adaptation Toolkit focus more on testing and verification of its outputs, the configuration files.

2.4. RISE for Traffica

RISE, short for Reference Information Service Environment, is by definition an application for creating, managing, editing and publishing operability data items within R&D development areas and towards customer documentation. RISE offers common data formats and enables creation and common storing place with a predefined, harmonized and agreed common process for metadata management. Thus it offers an agreed way of working. [RISE documentation]

RISE has been developed and maintained by a cooperation partner of NSN. Currently its development takes place in Poland. In this project the requirements for RISE for Traffica are specified by a NSN team located in Finland and the software development is performed by a Polish subcontractor team.

Figure 5 shows a typical view of how RISE looked like before current project. The image is from RISE Viewer which is used only to view the contents of RISE, but the GUI was unified all around the system.

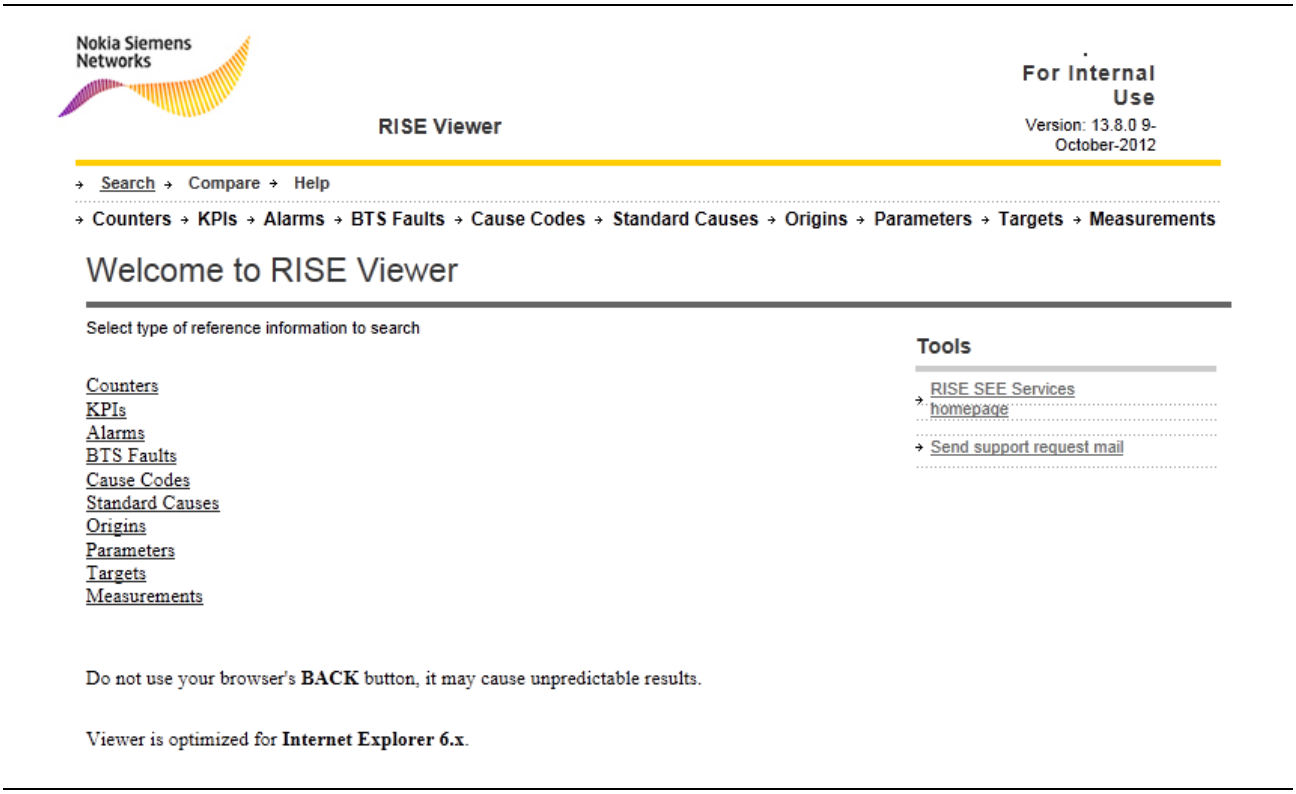


Figure 5. RISE's old and outdated appearance.

The problems with adaptation creation process were the leading cause to start looking for ways to improve the work flow. NSN has used RISE for long in other projects as well, so it seemed like a logical choice for the one place to store Traffica related information as well. Some new features would need to be specified and implemented in order to present and store the information correctly thus the project to create RISE for Traffica as a new section within RISE.

There are concrete benefits that using RISE could bring to adaptation creation process. Firstly it allows more automated testing and thus can save time from one day up to one week per adaptation. This helps to ensure time schedules are kept and gives more time for quality control – from formal reviews to avoiding identical naming and automated typing error corrections. The idea is also to have adaptation related documentation stored in the same place to act as a reference guide and allow automatic reference documentation generation which currently can take weeks to do manually.

2.5. Summary

Designing new tools for the work that has been done the same way for years is always challenging. In this case this is particularly true since the work done aims to change the whole workflow process. Developing RISE to be suitable for Traffica related work also means simultaneous work done on Adaptation Toolkit as their cooperation must be flawless in the new work process. One thing that helps is that these two tools share some of the technical requirements elicited from Traffica, only they approach them from different angles. Usability will be in major role in making separate tools meet and provide the user the experience of usefulness.

3. Usability

This chapter starts with discussions about the different definitions for usability. There are number of expressions what the term contains and this discussion attempts to find the common factors between them. Second part addresses the problems that are common when trying to implement usability into agile software projects. These considerations lead to discussion about theories on how to design and verify usability. In RISE for Traffica project the main goal for usability is to design an adaptation creation process that not only improves current workflow but is also accepted by its users. The concepts of usability and its role and specific challenges in this project are discussed in the last part of the chapter.

3.1. Defining usability

In order to determine a product's usability, or if its development work is increasing usability, it is necessary to define what usability is and how it can be measured. Usability as an attribute can be challenging to define and therefore many varying definitions exist. Commonly they all mention a product and a user. However, the definitions typically include a number of factors that can be used as the basis for measurements.

One the plainest definition is the summary presented in ISO 9241-11 (1998) standard, which defines usability as: "The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use."

The standard definition is a generalization but it gives keywords achieve, effectiveness, efficiency and satisfaction to go on with. Rubin and Chisnell [2008] employ the same terms when they define that "To be usable, a product or service should be useful, efficient, effective, satisfying, learnable, and accessible". Useful can be interpreted as something that helps a user in achieving his goals. The additions to the standard's definition are learnability and accessibility.

More comprehensive analysis on definition of usability is offered by Nielsen [1994a] who argues that usability is no single property that can be measured to define how usable something is. Instead he defines usability through five attributes:

- Learnability: The system should be easy to learn so that user can rapidly start getting some work done with the system
- Efficiency: The system should be efficient to use, so that once the user has learned the system, a high level of productivity is possible
- Memorability: The system should be easy to remember, so that the casual user is able to return to the system after some period of not having used it, without having to learn everything all over again

- Errors: The system should have a low error rate, so that users make few errors during the use of the system, and so that if they do make errors they can easily recover from them. Further, catastrophic errors must not occur
- Satisfaction: The system should be pleasant to use, so that users like using it

Nielsen continues that usability can easily be argued to be an abstract concept. These attributes, however, provide measurable variables that make usability a systematic discipline that can improve the process of product development.

While measurable attributes make usability a property that can be affected, Nielsen reminds it is still only a part of a bigger picture. In his model of system acceptability Nielsen states that the most important factor in deciding whether the system is good enough is does it satisfy all the needs and requirements of the users. In figure 6 Nielsen puts usability under usefulness, which in turn is under practical acceptability. These are the technical half of overall acceptability of a system: the other half being social acceptability.

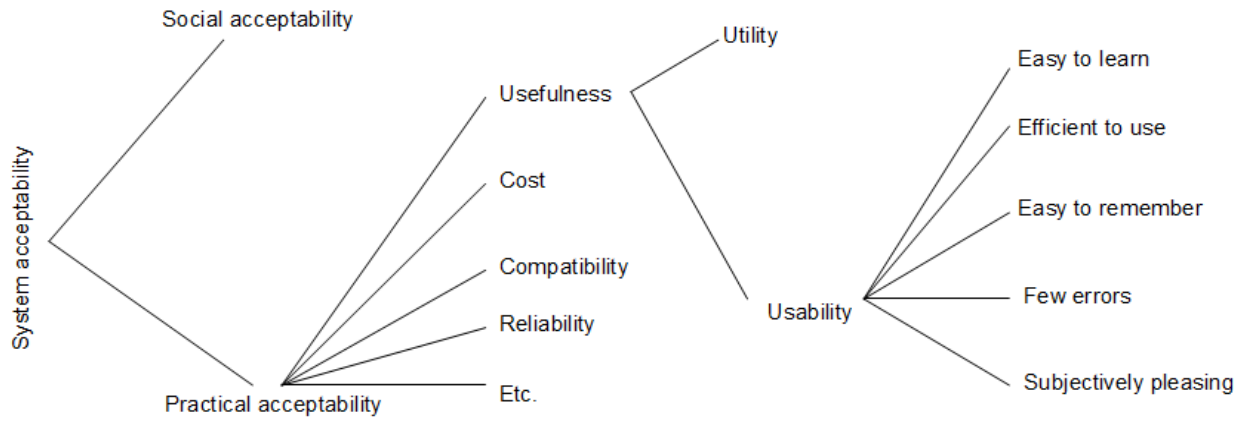


Figure 6. A model of the attributes of system acceptability according to Nielsen [1994a].

Nielsen does not downplay the role of usability but summarizes, that the most important question to keep in mind is whether the system is good enough to satisfy the users' needs [1994a]. The answer to this question decides the level of the whole system acceptability and ultimately plays such a big role that can cover for minor failures in usability.

3.2. Usability in agile software projects

Agile Manifesto [2001] had a huge impact on software engineering.

> "Individuals and interactions over processes and tools
> Working software over comprehensive documentation
> Customer collaboration over contract negotiation
> Responding to change over following a plan
>
> That is, while there is value in the items on
> the right, we value the items on the left more."

While transforming software development more iterative, it also affected usability design. Silva da Silva et al. [2011] performed a systematic review of existing literature on how user-centered design and agile software development methods have been integrated. They noticed that the major difficulty is keeping the Big Picture, when products are no longer thoroughly designed before their implementation starts. The review showed that usability work has developed towards iterative processes and the design and evaluation processes are repeated over and over again towards refined solution. The reviewers emphasize that the literature clearly states constant collaboration between designers and developers is very important factor in the success of software project, and that such methods should be selected that support this collaboration.

Constantine has been observing agile processes and outlined in his paper [2002] a simplified process to integrate usage-centered process to software development's lightweight methods. In his approach Constantine focuses on prioritization of tasks and creating and using prototypes. He also points out the same observation Silva da Silva et al. [2011] made about the problematic Big Picture of the design. Constantine notes that the up-front designing for user interfaces is often very minimal, but according to him must establish at least three things:

1) An overall organization for all the parts of the user interface that fit with the structure of user tasks
2) A versatile common scheme for navigation among all the parts
3) A visual interaction scheme that provides a consistent look-and-feel to support user tasks

McInerney and Maurer [2005] have also looked into differences and commonalities between agile methods and user-centered design in the same projects. They too have noticed that agile approaches prefer generalists and discourage extensive upfront design work. While this may be thought to cause problems, their study results have shown the exact opposite: all their gathered feedback was positive. McInerney and Maurer conclude that while seemingly different approaches, the specialized methods UCD provides for UI design can easily be employed in the iterative philosophy of agile methods.

3.3. Designing and verifying usability

Designing for usability is nowadays commonly called user experience (UX) or interaction design. While the differences between the two are largely academic, and in practice in corporate world they are used as synonyms the terms still have their own uses. UX design aims to design a subjective experience for users whereas interaction design drills more into essence of a product: what it is and what it does.

Cooper [2007] prefers the latter approach as he feels interaction design is more suited to address the main issue of designing how complex interactive systems behave. Cooper himself presents the Goal-Directed Design to help in creation of products that really meet the user needs. Goals in Cooper's terms are not tasks nor activities but rather meanings of them to the user. The key is to understand what motivates the users, what their expectations are and what they aim to achieve. Once a designer has an understanding of the users, he can create designs that are accurate and satisfy their needs.

While Nielsen advocates user testing as the best way to do proper product design, Cooper insists that thorough design phase is absolutely necessary and the key to satisfied users. Dubberly [2001] visualizes Cooper's process in figure 7, which also shows that Cooper does not play down the role of user tests, but in fact uses results from them as the fuel for tweaking the design.

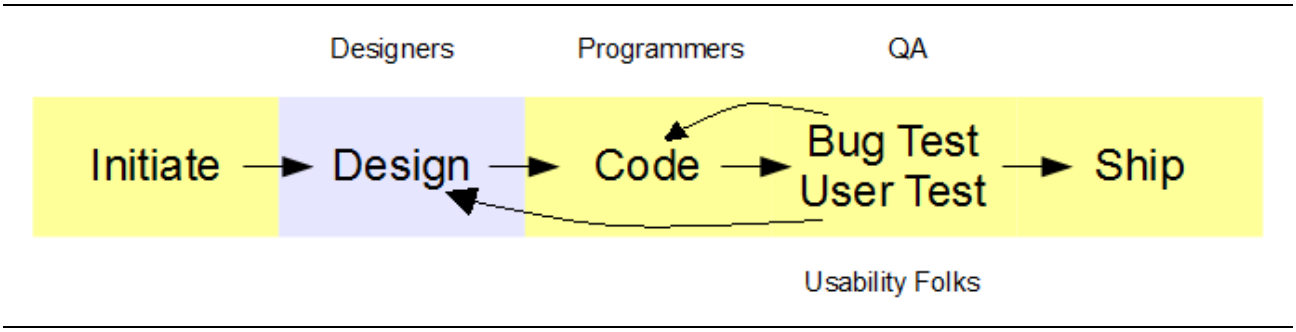


Figure 7. Initial design in the center of the Goal-Directed Design. [Dubberly, 2001]

The design process in Goal-Directed Design itself can be divided into six steps:

1) Research
2) Personas
3) Scenarios and Needs
4) Framework
5) Design Refinement
6) Development Support

Cooper's approach is not however entirely new idea. Gould and Lewis [1985] researched the topic of designing for usability already almost thirty years ago. The industry has changed a lot since, but their observations of using user feedback to the development of a product are still valid. The three designing principles Gould and Lewis suggested, early focus on users and tasks,

empirical measurement and iterative designs, are clearly at the base of the Cooper's model too. The Goal-Directed Design can be seen as having put these theories into practice and providing a detailed walkthrough to designing usability.

Verifying usability generally means user tests conducted in the late phases of product development. But as noticed above usability of the product can and should be tested along the development cycles, in iterations. According to Rubin and Chisnell [2008] using user testing as a verification tool for usability the measurements are two-fold: ensure the usability by measuring against the set usability goals, and confirm that previously discovered problems have been fixed and new ones have not appeared.

Rubin and Chisnell do not limit their vision of usability testing to assessment or summative tests. They emphasize that usability testing can be used as an explorative or formative study already quite early in the development. Their approach can be employed in the first steps of research and design in Goal-Directed Design. The user tests can for example use prototypes like paper protos to examine how effective the preliminary design concepts are. As to Rubin and Chisnell the added value from usability testing with unfinished or raw sketches of a product comes from the possibility to informal testing that allows the participant and moderator to work in collaboration and ask for ideas how everything should work.

3.4. The role and challenges of usability in the RISE for Traffica project

As discussed earlier that while usability can be defined as whole, it is necessary to define its desired level for each different project: what are the users' needs. It is also evident that the design process must walk hand in hand in collaboration with the software development and be adjustable to reiterations. Setting measurable goals helps to verify if further reiterations are needed, or has the product reached the desired level of usability. This chapter discusses how to employ these ideas in the development process of RISE for Traffica.

Adaptation creation is a complex process. Making changes to one step affects many others and thus the process need to be examined as one system. The current process has been around for awhile and all the user groups have become familiar with it. Now they are facing the situation where they are offered a new tool and with it a new way of working. Nielsen [1994a] talks about how, in situations like this, the role of usability is relatively small compared to the issue of system acceptability. He reminds that system's overall acceptability is the result of its social and practical acceptability. According to Nielsen the key factor is, is the system good enough to satisfy users' needs and requirements.

Kurvinen and others [Kurvinen et al., 2006] note that, based on their study on large projects, general usability design principles and guidelines do not always apply. They emphasize the context dependency of user-centricity and usability. These results are in line with basis of Nielsen's usability engineering and need to be considered in this project's scope: the developed system needs to have real relevance to its users in order to be accepted.

Maguire [2001] remarks that systems that users find usable and well designed tend to have improved user acceptance. While improved acceptance may be indirect result from proper usability design, his findings show that most users would rather use system that is easily accessible and easy to assimilate and use.

While total number of RISE users is around 5300 there are around fifty Traffica developers who will be using RISE for Traffica with few NE developers and some Customer Documentation (CuDo) specialists. This means that the implementation will be largely based on existing RISE. Cooper et al. [2007] suggest that "User interfaces should be based on user mental models rather than implementation models" which may prove to be hard to do on top of the existing system that is based on implementation models. The adaptations are also heavily depended on the structure they are built and not necessarily always on the idea how they are used. This leads to the conclusion that in order to enhance the system to meet the requirements set by Traffica an approach must be selected that aims for the minimum architectural changes with maximum benefits - the benefits being the desired result and the basis for usability goals.

The purpose of R4T requires for its users to understand the complex structure and connections within the stored data. This sets the requirement that its users cannot be assumed to be what Cooper et al. [2007] would describe as beginner-level. Cooper's division of user levels states that some users are always experts, some beginners but the biggest group is typically the intermediates. Since R4T is a tool that needs to feel efficient for its users, it cannot however require expert skills to learn to use it. Expert level users may still be offered alternative ways to operate it. While the system may not be designed for beginners, they still need to be taken into consideration because there are occasional personnel changes and due to the nature of the work sometimes some phases of it may come as new features to even experienced users.

As RISE is updated to meet the requirements set by the new Adaptation Toolkit its compatibility derives from Traffica and provides technical relevance. Implementing technical requirements, however, may not be enough. The main goal of the whole project is to enhance and optimize the adaptation creation process, so it is vital that RISE for Traffica not only meets the technical requirements, but also is accepted as useful and helpful by all its users. According to Maguire [2001] it can therefore be assumed that the role of usability is essential to this project's success: if the system is easy to use, yet efficient and useful in users' work they are more likely to adopt it into use. That is why the focus in designing usability should always be on the whole process instead of specific aspects or details of single pages or forms.

4. Usability methods

There are several methods available for ensuring usability of a product in development. Some can be described to be more of model of the process whereas some are more to the point direct approaches, single tools. The first group tends to have a set of steps that are followed through the development process. Each step then has its own recommended tools to be used at that particular stage of development. One typical example of these process models is ISO 13407 –standard based TRUMP (TRial Usability Maturity Process) that aims to model the entire product development process [TRUMP]. An example of direct approaches is the traditional usability testing of the product that can be employed at any stage of the process and even without larger framework of any process model.

Pihlajamäki [2010] called for more information about products' users and their needs in the same organization whose working practices this study aims to improve through implementing usability methods that best suit this line of work. Pihlajamäki suggests starting with easy to adopt methods like heuristic evaluation and paper prototyping, and then continuing with making usability issues more concrete and visible in the normal work of development teams. Even though not directly proposed in Pihlajamäki's paper, the process of creating personas to present users was partly motivated by his work.

Based on research question and direction pointed by Pihlajamäki this chapter examines and compares a number of different usability methods, both models and tools to discuss which of them would best suit the organization and finally select a method or combination of methods to be tried out in RISE4Traffica development process.

4.1. Discount usability

Discount usability is a usability engineering model that Jakob Nielsen proposed in 1989 [Nielsen, 1989] in order to simplify and lower the costs of developing usability. It is defined more by use of qualitative methods such as small scenario based tests with few participants and direct observations than quantitative and statistically established results of large and comprehensive usability testing. Nielsen based discount usability engineering on the use three main techniques:

- Simplified user testing with thinking aloud
- Narrowed-down or paper prototypes that support single scenarios
- Heuristic evaluation by inspecting interface design

Scenarios and heuristic evaluation will be discussed in more detail later in this paper.

Typical for all these techniques is that they can be employed quickly, at a low cost, with few participants and therefore are well suited for small tests of frequent iterations. The idea is that tests can be small and the product will be tested again after the results have been incorporated into the next iteration of the prototype.

Discount usability emphasizes early and rapid iterations with frequent usability input. This sets some requirements for the project and the organization. Rapid iterations are the spirit of currently prevalent agile methods and thus discount usability can be seen to be best suited for projects that employ Scrum, Kanban or any other such method. Given that the feedback from the designs is frequent the development team needs to be capable of quickly assimilating the results. On the organizational level it helps if the development team has constant access to, or has its own usability specialists. Also, the closer the collaboration between development team and users who participate in the test is the better.

Since it was first introduced, discount usability has been criticized for drawing conclusion from insufficient amount of data: only few users and quantitative results that are not statistically significant. Nielsen addresses this criticism [1994b] admitting that for much research the high degree of confidence is required, but for more pragmatic tasks like developing usable interfaces a less formal approach is often satisfactory. In his words: “In discount usability engineering we don't aim at perfection anyway, we just want to find most of the usability problems.”

Others have been eager to utilize Nielsen’s simplified approach too. Sohaib and Khan [2011] explore the integration of discount usability into extreme programming. Their findings show that especially with the rise of agile methods and iterative design approach, where requirements elicitation continues throughout the project, the ability to utilize usability techniques fast at any time of the project is the ideal solution. Therefore discount usability fits the need for model that helps to integrate usability techniques into programming.

4.2. RITE

RITE, Rapid Iterative Testing and Evaluation, is an iterative usability method that was introduced by Medlock at al. [2002]. It is characterized by its fast response to identified usability problems. RITE can be said to be a further developed method of Nielsen’s discount usability: the basis of finding usability issues with few, or even only one, participants and lightweight methods is the same, but after discovery RITE focuses on finding solutions to and fixing the problems in the shortest possible time. Typically for RITE two participants do not test the same version, but the second participant tests the version fixed according to findings from the test with the first participant.

Rite is developed with the business reasons of usability testing in mind; speed and efficiency are the controlling factors. Trying to come up with efficient usability method for corporate use Medlock et al. [2002] identified and tackled four problems in their effort that lead to the development of RITE:

- Decision makers do not believe usability problems are worth fixing
- Scarce resources typically favor adding new features over fixing ones that work somehow
- Usability feedback arrives too late to be useful in design phases
- Development teams' unwillingness to spend time on tasks that are not verified to fix the problem

RITE developers did not want to re-invent the wheel, so the method is very similar to traditional usability testing tuned with Nielsen's ideas about discount usability. As mentioned the process of getting results from tests does not much differ, but the biggest change is the rapidity of handling them. In RITE as soon as a problem has been identified and its solution cleared, the changes are implemented to the user interface. This can happen even after testing only one participant. Traditional usability testing would take 5-7 participants and even Nielsen suggests three before any changes are applied. Of course in using RITE this is not followed blindly, but the issues are also classified. The classification of found issues in RITE somewhat differs naturally, as some issues are obvious as are their solutions, whereas some issues may be due other factors and require further data collection.

Using RITE sets some requirements both for the testing and the development team. The importance of issues is based on prior to testing agreement on which tasks every user needs to be able to complete. After each testing the hindering issues are evaluated on their importance and classified. This then determines the course of action that follows as some of the needed changes will be started to implement right away. This means that the development team must have time and resources available at that designated time.

4.3. Usability testing

Usability testing may refer to the use of any of large amount of methods and techniques that are used to measure or evaluate the usability of a product or system. Here it is used to refer, the same way as Rubin and Chisnell use the term [2008], to the process where representatives of target group as participants test how a specific product meets its usability criteria.

Usability testing in its many forms is probably the most commonly used usability method. If the correct testing method is selected, it can be very efficient in discovering usability problems. Alan Cooper [Cooper et al., 2007], the advocate for Goal-directed design, states that usability work is more than just testing, but admits that usability testing is especially effective in determining things like:

- Does the section/button/label naming make sense?
- Is the information grouped into meaningful categories?
- Are the common items easy to find?
- Are the instructions clear, or necessary?
- Can the tasks be completed efficiently?
- Do users take missteps, and if, where and how often?

While Cooper at al. [2007] may mainly focus on evaluating the first-time use of a product, their list still matches Nielsen's [1994a] 5 usability attributes to look for in a usability test:

- Learnability: How easy it is for users to accomplish basic tasks the first time they encounter the design?
- Efficiency: Once users have learned the design, how quickly can they perform tasks?
- Memorability: When users return to the design after a period of not using it, how easily can they re-establish proficiency?
- Errors: How many errors do users make, how severe are these errors, and how easily can they recover from the errors?
- Satisfaction: How pleasant is it to use the design?

The point that Cooper et al. [2007] emphasize is that usability testing should be iterative and the test types should change throughout the lifecycle. At first testing should be exploratory, then proceed to assessment of features and finally to verification. Rubin and Chisnell [2008] agree with Cooper's point of view, but also bring forward the thought that there should always be some goal for the testing. Their examples include informing design, eliminating design problems and frustration and improving profitability. The set goal then determines on which attributes from above lists the test will focus.

Rubin and Chisnell also present the sets of basic elements for usability testing and the limitations of testing. Their basic elements are:

- Development of research questions or test objectives rather than hypotheses
- Use of representative sample of end users which may or may not be randomly chosen
- Representation off the actual work environment
- Observation of end users who either use or review a representation of the product
- Controlled and sometimes extensive interviewing and probing of the participants by the test moderator
- Collection of quantitative and qualitative performance and preference measures
- Recommendation of improvement to the design of the product

The limitations that they present to the testing, why it does not necessarily guarantee with one hundred per cent certainty that product is usable:

- Testing is always an artificial situation
- Test results do not prove that a product works
- Participants are rarely fully representative of the target population
- Testing is not always the best technique to use

Cooper, Nielsen and Rubin set up a good model for planning usability testing. They each have gone into more details on what, how and when to test, but the above checklists are a good start. The important things are to remember the target group, context, product itself and the phase of development process and select the approach accordingly. The correct combination with proper preparation is the key to getting helpful results, but it must be remembered that no preparation ever guarantees a successful outcome.

4.4. Heuristic evaluation

Heuristic evaluation is a usability inspection method to help identify and evaluate usability issues in designing user interfaces. It is an informal usability analysis method introduced by Nielsen and Molich [1990] who observed the need to compose common guidelines to one of the most used ways to analyze a user interface: looking at it. In heuristic evaluation a group of evaluators are tasked to look at a user interface design and then comment on it. This is a lightweight usability method that can be conducted without users.

There can be any number of evaluators, but practice has shown the optimum number to be between three and five. Single evaluator can be used, but it is not recommended because different evaluators typically find different problems, and an essential part of the evaluation is to compare comments between evaluators. The evaluators do not need to be usability experts but they need to have knowledge on the principles, the heuristics, which the evaluation is based on.

Common practice to perform a heuristic evaluation on a product is to have each of the evaluators do it independently of each other. After the evaluations, the evaluators gather together to go through their findings and discuss them. This phase produces the severity classification for discovered problems and a summary of the evaluation. Depending on the ways of working the summary can be reported or delivered verbally. Whatever the method, the most important findings should be emphasized.

Nielsen's [1994c] list of ten usability heuristics is likely the best known and most widely used set of heuristics employed due their general nature, but it is not the only one. Others include for example Gerhardt-Powals' cognitive engineering principles [Gerhardt-Powals, 1996], which are another well-known, more research-focused set of heuristics, and Connell and Hammond's 30 usability principles, that drill deeper into details of human-computer interaction [Connell 2000].

Heuristic evaluation is inexpensive and quick method that can be used at any phase of the development process starting from the early sketches. If assigned to correct heuristic it can be easy to suggest corrections to the discovered problems. Method is also easy to learn and understand which helps motivate people to do it and it does not require much advance planning. [Nielsen and Molich, 1990]

Heuristic evaluation has been sometimes criticized for that even if it is an easy method to apply it still requires some knowledge and experience to use heuristics effectively. This may lead to a situation where heuristic evaluation is conducted by a group of usability experts, which in turn can become expensive. Another criticism towards heuristic evaluation is that it has a propensity to identify low-severity minor issues that are not real problems and not discover the major usability problems.

4.5. Personas

In order to design for end users, the designer needs to have understanding of them. The understanding comes from information and there are many ways to collect it. The problem designers face is that using wide variety of methods to gather different user information leaves them with surplus of scattered data. To filter the essential information and communicate it efficiently a model is needed.

Cooper [1999] introduced and later refined [Cooper at el. 2007] the concept of persona to the HCI-field to create models to describe how users behave, how they think, what they wish to accomplish and why. Personas are never actual people but combined archetypes based on the observed behaviors and motivations of real people. Personas are precise descriptions of characteristics of these people and what they want to accomplish. Preciseness is important as the personas represent the actual users throughout the whole designing process.

A persona is never a random model, but always based on research. Each persona represents one important group of target users, but is represented as an individual, a persona as the name suggests. Every persona has its own name, photo, background, personal and professional goals and motivations and typically a slogan expressing his/her personality and motivation. Cooper's model to create and use personas focuses on goals and designing action scenarios. User goals are user motivations, and commonly inferred from qualitative data. Cooper uses the term Goal-directed design (GDD) to describe how the user motivations motivate usage patterns, become design goals and are communicated to development.

In many cases it is possible to identify the group of people who will be the main user group of the product. The persona created to represent this group is called primary persona and it is the one whose needs weigh the most. A secondary persona is often created to supplement the primary persona, to present the second most important group of users. More than two personas can be created if there is need for them, but in most cases the two personas cover the ground quite well.

Using personas in communication is one of their advantages. For example software developers often have poor understanding of the users and easily assume they are similar to themselves. This is why personas need to feel like real persons: the more real they feel, the easier they are to communicate.

Pruitt and Grudin [2003] suggest in their results that one reason for creating personas is to start discussing products or their features in terms of the personas. In such a case personas would answer to questions like "Why are we building this feature?" and "Why are we building it like this?" The writers also ponder whether Cooper's method is fit for all cases and if its strength lies in support of other methods and not in replacing them. Pruitt and Grudin outline a psychological theory to enhance the use of personas which suggests that personas work best if their creation is an iterative process and they are developed for particular effort. They conclude that personas are most valuable as a means of communication to all stakeholders and not just developers.

An interesting viewpoint to iterative creation of personas comes from Wolkerstorfer et al. [2008] who in their study of integrating extreme programming to HCI present the concept of extreme personas. This approach applies small iterative steps to personas to refine them, but has the readiness to refactor or extend personas if significant new insights will be developed for them.

The major advantage of personas comes in the form of communication and finding common consensus as Chang's [Chang et al. 2008] results confirm. They are relatively easy to use and stakeholders tend to have good understanding of what they represent. The biggest risk related to personas is getting them right. Both Chang and Pruitt and Grudin [2003] acknowledge that personas may not always be based on actual research on users, but they may reflect designers' own thoughts and experiences. Another possible risk is that information that does not fit into personas gets filtered out and will not be used in future development or selecting participants to usability tests.

4.6. Scenarios

Scenarios are a design tool that connects research data to design solutions. They exist to define what the product should do before designing how it will do it. Scenarios derive from personas and use them to tell a short story how a specific user (persona) achieves specific goal. The purpose of the scenarios is to clarify to designers and developers what the product needs to be capable of doing. Product's data objects and features are distilled from detailed scenario and implemented into a design solution that is developed to match scenario's description.

Cooper [Cooper et al. 2007] states that scenarios provide four aspects to solution design:

- Scenarios are presented as stories to convey the image of ideal user interactions
- They are used to define requirements
- Based on requirements, scenarios in turn define the fundamental interaction framework for the product
- All the aspects are kept together by narrative that uses personas to create stories that point to desired design

Cooper [Cooper et al. 2007] also presents three types of scenarios. The first type is a context scenario, which is written from a persona's point of view and focuses on that persona's activities, perceptions and goals. Context scenarios provide the idea of what the user does and what she wants to accomplish with this product. They form the basis for design and are thus created before any design is done.

The second scenario type is a key path scenario. They are revised versions of context scenarios and aim to describe the interactions between user and the product more precisely. Key path scenarios pick the most important user tasks, keep focus on the persona and her goals, and attempt to model each step of the task progression as accurately as possible. Key path scenarios typically go through many iteration rounds as more details develop along the way.

The third type of scenarios is a validation scenario. These scenarios typically do not include great amount of details as they are mainly intended for testing purposes. Validation scenarios', as their name suggests, purpose is to test design solutions and how they work in varying situations. Results from using validation scenarios can be employed to further iterate key path scenarios.

Scenarios are in many ways similar to use cases: they both describe the interaction steps user has to do in order to achieve specific goal. Use cases are however more concrete and technical in nature describing the behavior of the system and focus on low-level user actions. Scenarios are more human-centered. Scenarios also prioritize the system's functions and the way they are expressed to users in terms of perception and interaction.

4.7. Discussion about selected methods

RISE for Traffica is a typical agile software project in that it is specified and implemented in iterations and its features may change as the work progress. When the iteration cycles are short the usability approach needs to be able to provide concrete suggestions, repairs, fixes and updates for evaluation or design – suggestions that can be conveyed to developers as such to be implemented or fixed. In short the ideal approach would be fast and inexpensive and producing implementable solutions. This kind of lightweight method would also match the suggestions Pihlajamäki [2010] made to enhance the usability awareness in the same organization: use easy methods and spread the knowledge of usability in the process.

Discount usability and RITE, as discussed above, are not usability methods as such but more of framework models to guide in the selection for used methods. Discount usability emphasizes more the thorough, yet simplified, user testing. RITE on the other hand somewhat criticizes usability testing for focusing too much on problems instead of coming up with rapid solutions and dealing with them. They both do endorse user testing as long as it is simplified. The two approaches are not that different in the end and they could be integrated into one mindset to be employed with R4T project.

Such mindset would require usability tests that could be performed even with only one participant. The effectiveness of the tests would be measured by the results: how many concrete

action points could be derived from the test. The tests would be conducted regularly, whenever there were new features available. The usability tests would always concentrate on new features and verifying the fixes to previously discovered problems.

Because the implementation of new features can take time and increase the interval between the tests, supplemental methods would be required. Some of the tests could be conducted on paper prototypes of the system. Also heuristic evaluation could prove to be very useful when used to complement the results from the test with users – never replace them. The main target for heuristic evaluation could be inspecting interface design as was suggested by Nielsen [1989].

Planned implementations need to be designed first in order to get feedback from them. Neither Discount usability nor RITE addresses designing methods directly. However the personas and scenarios form the basis of Cooper's goal-directed design [Cooper et al., 2007]. These are methods that can be used to model the users of the system and how will they use it. This is the information that tells what is required of the system and therefore the very base of designing.

Although not in the mindset of fast results the goal-directed design can be employed along with it. The research process for personas and scenarios is extensive, but iterative. While the beginning of the implementation for R4T follows technical specifications, it gives time to work on the research. The implementation can be tested from the first versions on and directed to fix the discovered usability problems. The solutions to these usability problems should be based on the work that has been going on for creating personas and scenarios.

4.8. Summary

The usability work for RISE for Traffica has two targets: evaluate current solutions and provide improvement suggestions in short iterative cycles. That means usability testing will be in a major role and due to time and resource constraints, the focus in it will be on discount usability methods: usability tests with single participants and heuristic evaluations. In designs the focus point will be on basing new designs on users' needs and goals derived from personas and scenarios. Cooper's goal-directed design and other eligible approaches can be used as reference and employed where possible.

5. Results from usability testing and personas creation

During this study altogether five separate usability tests were conducted on RISE for Traffica. In this chapter the tests and their results are presented in chronological order. The used methods plus how they were developed and employed are also described. Each test also has a short discussion about its findings and conclusions. Even though the personas and scenarios were out of the scope of this study, the initial work done on them is presented here as a basis for further development.

5.1. Usability tests

The purpose of the usability testing was primarily to discover usability problems in the development version that was current at the time of the test. Since this was work in progress, it was expected that technical problems would surface as well, and these would be reported too. Another goal for the testing was to get feedback from the primary target group that would be using the system once ready. The user feedback was gathered, analyzed and developed into enhancement ideas. The discovered problems were also analyzed and prioritized based on their occurrence frequency and severity (Table 1).

	SEVERITY			
OCCURRENCE	**Low**	**Medium**	**Serious**	**Critical**
Rarely	Low	Low	Medium	Medium
Sometimes	Low	Medium	Medium	High
Often	Medium	Medium	High	High

Table 1. The prioritization of problems.

Discovered problems were grouped and put in order according to their severity. The severity levels of usability problems were:

1) Critical: Prevents users from using the product the intended way. Need to be fixed urgently.
2) Serious: Significantly complicates completing common tasks. Fix as soon as possible.
3) Medium: Complicates the use of product to some extent and frustrates user, but does not affect task completion. Need to be fixed.
4) Low: A quality or cosmetic problem. User receives negative and unfinished image of the product.

T) Technical problem: Includes missing features.

C) Content related issue: Logical error, or otherwise erroneous content. May cause usability problems indirectly.

The prioritization scale was low, medium and high. The high priority issues were to be handled first. The handling of issues was further divided into three categories:

Fix	A repair to a certain problem, action point for developers
Study	Needs further research, action point for designers
New	New, previously unaccounted for issue or feature, action point for both

All the participants for the tests were picked up from within NSN personnel, among Traffica developers (Table 2). The heuristic evaluation had no participants. All the tests were conducted in meeting rooms in company premises unless otherwise mentioned.

Test	Number of participants	Sex	Nationality	Age	Expertise in Traffica (years)
1st	1	male	Finnish	40-45	10+
2nd	1	male	Finnish	35-40	10+
3rd	2	male / male	Finnish / Finnish	45-50 / 35-40	10+ / 1-3
5th	1	male	Indian	30-35	5-10

Table 2. The demographic of test participants

5.2. The first two usability tests for RISE

The first two usability test conducted on R4T followed the traditional pattern of having user, unfamiliar with the system, from target group trying to do given tasks. The tests started with a warm-up task that was intended to reduce the possible stress caused by the test situation and also to give first impression of the system in question. Both tests had six tasks and their completion was observed. Task completion was classified as: Task was successfully completed, Test moderator helped participant, Task failed, Task was aborted or Task was skipped. No timing was used.

5.2.1. The first test

The purpose of this test case was to test the usability of the first implemented features for the renewed RISE. The test was conducted with one participant in November 2011. The participant was male adaptation specialist with several years of experience with adaptation creation, who used RISE test environment to do a number of tasks. The tasks required user to fill in a number of forms. The user was provided with necessary data to do this. The results from the test completion are collected in the Table 3.

	Task	Nr. of problems	Task completion
1.	Create new adaptation with given data.	3	successful
2.	Modify the adaptation's description.	1	successful
3.	Create a new RTT-report with given data.	5	successful
4.	Add given fields to RTT-report.	9	failed, aborted
5.	Rearrange RTT-reports' new fields.	-	skipped
6.	Logout from RISE.	-	skipped

Table 3. Tasks and results from the first usability test.

From the first test total of 16 usability problems were reported, when some problems were recurring. Problems' severity ranged from critical to low, and as was expected: technical and content related problems were discovered as well. The total findings from the test were:

- 16 usability problems
- 16 content related problems
- 7 other issues that required further action
- 4 positive discoveries

Of these discoveries 25 action points were issued:

Priority	Type	Number
High	Fix	5
	New	2
Medium	Fix	2
	New	6
	Study	3
Low	Fix	4
	New	1
	Study	2

Table 4. Division of action points from the first test.

5.2.2. The second test

The second test was executed similarly to the first with one participant, another adaptation specialist, who used RISE test environment to do the given tasks. Location was a meeting room with the participant, moderator and observers present. Again the tasks required filling a number of forms and the participant was provided with data to do this.

	Task	Nr. of problems	Task completion
1.	Create new NE adaptation with given data.	3	successful
2.	Add RTT report with given data to the just created NE adaptation.	7	successful
3.	Add new version of RTT report with given data to the just created NE adaptation.	4	successful
4.	Start creating new RTT report and add three standard fields to it.	5	helped, failed
5.	Rearrange the order of the fields you just added: Make the third the second and the first the third.	-	skipped
6.	Add two new Recognition rules with given data.	5	failed

Table 5. Tasks and results from the second usability test.

From the second test total of 21 usability problems were reported, when again some problems were recurring. Problems' severity ranged from serious to low, and again some technical and content related problems were discovered as well. The total findings from the test were:

- 21 usability problems
- 8 content related problems
- 11 other issues that required further action
- 5 positive discoveries

Of these discoveries 23 action points were issued:

Priority	Type	Number
High	Fix	2
Medium	Fix	6
	Study	3
Low	Fix	6
	New	3
	Study	3

Table 6. Division of action points from the second test.

5.2.3. The combined results

In the first tests for R4T the two participants tried to complete the total of nine different tasks. Out of these nine tasks, seven were successfully completed. The cases that required help from the moderator or were classified as failed, did not fail because of critical usability problems. In all failed cases the common factor was yet to be or only partly implemented features, which distracted the participants. Relatively high percentage of tasks, three in all, was skipped mainly because of time running out. One task was skipped because the participant had run into and commented on that feature in an earlier task.

The findings from the tests were many. Total of 79 issues were discovered (Table 7). Out of these 79, 37 were identifiable as unique usability problems, 24 were categorized as content-related problems and 18 were issues that typically pointed out the need for further studies on the subject.

Total	Usability	Content	Others
79	37	24	18

Table 7. The results.

Table 8 presents the division of the 37 usability problems. In all three critical usability problems were discovered, four serious, eleven medium and seven low level problems. Some technical and content related problems were classified as usability problems because the main problem was not with the features that would not work, but the logically erroneous ways they were implemented.

Severity / Test	Critical	Serious	Medium	Low	Technical / Content
1st	2	3	5	1	5
2nd	1	1	6	6	7
Total	3	4	11	7	12

Table 8. Usability problems by severity and by test.

Content-related problems and other issues formed half of the findings. There was no typical case for these. Examples of gathered notes include for example the system asking the input in strange order or format, participants not understanding the used terms or expressions, remarks that some of the content could be generated automatically based on user's previous input, unified naming styles or even whether to use pop-up dialogs or not.

5.2.4. Discussion about the results

Both of the first two usability tests for RISE for Traffica followed the pattern of traditional usability testing where user is given one task at the time and then observed as he tries to complete it. There was only one participant for each of tests, which is not much. It is however important to keep in mind that both the tests were conducted on very early versions of the system. The tasks also presented typical use cases from adaptation creation process as was confirmed by the participants. Although RISE for Traffica presents a new way to work with adaptation data, both the participants, who were experts of many years and deep understanding on the subject, seemed to pick up the idea quite fast.

It must be admitted that especially the terminology used at times confused the participants. This lead to very specific questions about the details of the system, which in turn turned out to provide very useful insights towards the development of the system. Another distracting factor in the tests was the missing or at the time only partly implemented features. It was clear from the start that the participants were interested in the system and wanted to explore it more than was required to complete the tasks – or indeed even more than was implemented of the system's features. This behavior, similarly in both the tests, lead to time running out and consequently skipping of few tasks.

As the target of these tests was to gain more information for further design and development of R4T and not only pick up points for usability improvements, the discussions that followed from the participants straying off the tasks proved very advantageous. These comments were added under content related and other issues, and it is notable, that they formed almost a third of the usability problems plus over half of the total findings recorded from the two tests. The user comments were especially useful in pointing out subjects that required further studies on how to refine the logic of the system.

Comparing the results from the tests showed that they both resulted in similar number of findings. In the first test the content related issues had clearly higher percentage than in the second test. The second test in turn revealed more usability problems. It is also notable, that the number of critical and serious usability problems was reduced from the first test to the second. This suggests that turning the findings into distinct and prioritized action points managed to convey the test results into development tasks.

Some of the findings and feedback were the same between the two tests. Although the results were promising, there were signs that the traditional usability testing can only provide limited amount of information as Rubin and Chisnell [2008] had suggested. The clear indicator was the discussions that seemed to reveal more than just observing participant actions and talking aloud: participants had many questions, comments and good ideas on how to do things. On the other hand, the results were not altogether the same which suggested that the employment of RITE method was providing useful results. Based on the analysis of these tests it was decided that further testing needed to focus more on interaction between the test group and participant and not just observations.

5.3. The third usability test for RISE: An expert walkthrough

The purpose of this test case was to perform a type of expert walkthrough and evaluation to current test environment of RISE. The focus of this test was to run a scenario that simulates the normal tasks an adaptation specialist performs with the system. Previous tests gave parallel results plus useful information on technical aspects of RISE and its feasibility in adaptation creation. To dig deeper into this area and get more data on areas that a traditional usability testing on work-in-progress product might not reveal, a different approach was selected for this third usability testing.

5.3.1. Development of test method

The idea was to get more feedback through discussions while the participants were working on given tasks. This walkthrough combined elements of cognitive walkthrough, contextual task analysis and participatory design. The idea was to go through, one step at the time, the most typical use cases R4T supports. The point of interest was to ask the following questions at each step:

1) Will the user try to achieve the right effect?
2) Will the user notice that the correct action is available?
3) Will the user associate the correct action with the effect to be achieved?
4) If the correct action is performed, will the user see that progress is being made toward solution of the task? [Cognitive walkthroughs]

From contextual task analysis the sequence model was employed to observe the procedural steps the user takes in order to reach his goal. Here the user was let to decide the course of action and comment on how logical or intuitive the sequence of actions was.

Participatory design asks the user questions about how to solve design problems. In this case the work was in-progress, so in case of incomplete or obscure features or other problems, the user

could be asked to share his perception of the problem or required procedure and even to provide his own unique ideas and solutions.

The assumption in utilizing a work-domain expert as an evaluator rather than just a user was to gain more insight into problems and challenges users face with their work. This way the approach goes beyond aspects of usability and collects user experience issues as well [Expert walkthroughs].

5.3.2. Test results

The test was conducted with two participants (Table 2), both of them adaptation specialists. One of them was more experienced in this field than the other - over ten years of experience on Traffica adaptations compared to one - and the idea was to have this senior participant to explain the other what was done and how while inputting adaptation data to RISE. Location was an office meeting room with the participants, moderator and observers present. This test was to have additional observers online, but that plan was dropped.

The test consisted of going through RISE for Traffica test environment's latest implemented features, such as CCDs Possible values and Data types, and discussing about them. The main focus was to observe if the correctness and logics of action sequences were implemented properly. A warm-up task (Table 9) was presented as an easy approach to RISE. Task 1 was then basically the rest of the test. This time no additional adaptation data or interface specification was provided, since participants had extensive knowledge over NE adaptation which was used as input.

Warm-up task: Let's just create a test Traffica release for this test. Start condition: Logged in to RISE on Traffica Main Page. End condition: New Traffica release has been created.
Task 1: Continue as you would fill in new Traffica. We'll go through views one at a time and may discuss them. You may ask questions, or we may ask about your views and opinions. Start condition: Traffica release view with created release. End condition: We have gone through the most of the views.

Table 9. Test tasks for the third R4T usability test.

Total number of findings from the test was 28 (Table 10). Out of these fourteen were usability problems. There was the same number of content-related and other findings, which were mostly ideas about how some existing features could be improved or done differently. Few were completely new ideas.

Total	Usability problems	Content-related problems	Other issues
28	14	3	11

Table 10. The number and division of problems.

Table 11 presents the division of the fourteen usability problems. In all no critical usability problems were discovered, two were serious, three medium and seven low level problems. And as with the first two tests, some technical and content related problems were classified as usability problems as they were confusing to participants.

Critical	Serious	Medium	Low	Technical / Content
-	2	3	7	2

Table 11. Usability problems by severity.

Content-related problems and other issues formed half of the findings in this test too. Typically they were about technical details from the need to generate some of the content automatically to layout details, visibility and ways to navigate within R4T. Some issues still had to do with used terms.

Based on this test's results, total of twelve action points were issued (Table 12).

Priority	Type	Number
High	Fix	3
Medium	Fix	1
	New	1
	Study	1
Low	Fix	2
	Study	4

Table 12. Division of action points from the third test.

5.3.3. Discussion about the results

After the first two usability tests on RISE for Traffica gave parallel results it was decided to try something different. It was deemed that the qualitative feedback on system that had been gathered was especially valuable as it could easily be transformed into action points for development. More

of the conversational feedback was the target when developing the expert walkthrough –approach to the third usability test.

The system had developed from previous tests and was in state to allow some of the phases of the adaptation creation process to be performed. Using two participants as mentor and protégé was a last minute decision before the test. It is however a typical approach to how the adaptation specialists are trained to their work so the idea was justifiable. And indeed in this case the two had been working together in this sense. They knew the adaptation they were working on well, but the system, R4T, was unfamiliar to both of them.

The conversational method was new to everyone and it did not start off quite as well as imagined. The participants were reserved about the situation and the new system they were introduced to. At first the new approach to adaptation creation had to be explained and it took awhile for the participants to get hold of the system: which functionality represented which step of the process. The start was slow and discussion mainly consisted of questions and wonderings. Little by little, however, the pace quickened as the participants got more familiar with the system and dared to try different things with it. This opened up the conversation more and shifted it towards the work flow and its logics. By the end of the test the participants were excited about R4T as one comment clearly revealed: *"Why haven't we had this before?"*

Comparing the results to the two previous tests show that this test resulted in fewer findings. Their division was however similar as half of them were usability problems and the other half consisted of content-related and other issues. The severity of discovered usability problems was clearly milder than before. This can be explained by the development of the system and the features that were more completed than on earlier rounds. This also shows that transforming previously discovered problems into distinct action points for implementation paid off, as the biggest problems were fixed.

While usability problems were milder than before the rest of the findings were also different. In the first two tests the content-related issues were emphasized, but their number reduced to this third test. The feedback from the actual users had also been managed to convey into improvements. The other issues in the test were mostly collected ideas on how the views of the system and the workflow could be outlined in alternative ways. There were very few technical problems.

Although the third test was not a complete success it showed that direction was correct and adopting new ideas from cognitive and expert walkthrough theories added to the value of results from usability test. Getting concrete action points from the test results had proven to be very useful feature as they had been conveyed further into actual improvements of the product. Some more action points were also made from this test. The test format itself still needed some improvements. The expert walkthrough method seemed promising, but it still needed tweaking.

5.4. The fourth usability test for RISE: A heuristic evaluation

After three usability tests it was time to evaluate RISE for Traffica from another viewpoint and therefore a heuristic evaluation was performed. This test was conducted by two usability experts.

5.4.1. The test background

The used method was to go through current implementation of RISE test environment view by view, simulating the actions of an end user. Each view and its functionalities were evaluated according to Nielsen's heuristics [Nielsen 1994b]. The purpose of the evaluation was to discover and identify possible usability problems. The problem, in this case, being an instance where one or more of used heuristics was breached.

5.4.2. Test results

The heuristic evaluation resulted in total of 30 problems discovered. Some of them were repeating, like certain design patterns, and were reported only once. The division of findings based on their severity is presented in Table 13. No critical problems were encountered and only one serious usability problem. Five of the identified problems were categorized medium-level and one of them was also in connection to the definition of content. Largest number of problems, thirteen, were low-level usability problems. Three of these also had connections to the problems with content specification. Eleven of the findings were directly categorized as being content-related. No technical problems were run into during the test.

Severity	**Critical**	**Serious**	**Medium**	**Low**	**Technical / Content**	**Total**
Usability	-	1	5	13	11	30
Also content-related	-	-	1	3	n/a	4

Table 13. Results from the heuristic evaluation.

Each problem description came with a repair suggestion. These were meant as a starting point for further design plans. The only action points issued were study and fix items for the content.

5.4.3. Discussion about the results

The first heuristic evaluation for R4T was a quick look on the system from usability expert point of view. The test was planned as a complementary to the third usability test as they were performed within very short period of time: the fixes from the usability test had not yet been implemented but some wire frame models were available for new to-be-implemented features. Overlapping findings

between the two tests were not included in heuristic findings unless heuristic evaluation revealed something new about them.

The findings of heuristic evaluations are typically reported including mention of breached heuristics, but in this case the focus was in quickly identifying potential problems and coming up with a repair suggestion and therefore breached heuristics were listed during the process, but not reported.

The results from this evaluation were in line with the previous tests in that there were very few if any critical and serious usability problems, but most were low level. These problems had mostly to do with cosmetic or quality issues, or were unclear instructions. These can be put under work-in-progress category as some of them, like the tool tips, had barely just been implemented and their content and usage were not yet determined. Medium level usability problems were all related to too complicated or illogical ways of doing things, or then there were same functionalities that were implemented differently. Special attention was paid to come up with suggestions that would work as general solution.

After the drop in the number of content-related problems in previous test the number of them rose again in this one. Since the system was basically the same it can be assumed that the participants in the previous usability test did not quite make it to the level of deeply analyzing the content they were working on, but were preoccupied by the more visible aspect of the new work environment. The other usability expert participating in this evaluation had background also from working with adaptations and could therefore point out most of these issues with content.

The reducing number of serious usability problems showed that the progress was going in the right direction: picking out the first major usability problems worked as a guideline for the developers. The rise of lower level usability problems was a natural reaction to that progress, as more functionalities got implemented the GUI was starting to refine and smaller details were starting to need tweaking.

The content-related problems showed that while extensive work was put into requirements specification it was not yet perfect. The challenge came from the new way working, which could not really be tested before some of its work phases were implemented. Since the content issues were included in the usability test plan for R4T these results could be gathered and conveyed to further specifications.

One factor that can be contemplated in possible future evaluations is the selection of used heuristics. As mentioned in the theory section there are more than just Nielsen’s heuristics available and doing a comparison between two heuristic evaluations that were performed on the same system but with different set of heuristics could be enlightening.

5.5. The fifth usability test for RISE: a remote expert walkthrough

The fifth usability test for RISE for Traffica was an improved version of previous expert walkthrough test. The improvements included even more focus on interaction between the participant and the test group. This test was also the first to be conducted remotely which added some challenge to the execution.

5.5.1. The test background

Increased interaction during the test was aimed to achieve through specifically assigned roles for the testing group. The moderator's role remained largely the same but the major improvement was to design roles for Big Picture Thinker and Technical Communicator and train people for them. Big Picture Thinker was someone with comprehensive knowledge and skills to ask about or answer to questions about the whole work process. Technical Communicator's role was to answer to questions about technical and content-related issues as well as to prepare to ask about the development team's study items and possibly other problematic issues that would surface during the test.

Another major change to test execution was that it was arranged remotely. This was a new feature for everyone involved. Test was arranged over TelCo and WebEx-remote connections between Finland and India. Test was arranged with one participant, an experienced adaptation specialist. The participant had remote control over test moderator's computer and performed test tasks on it. TelCo was used for communications while WebEx and webcam were used to share the view of the desktop and participant/test group. Test group had assembled in a meeting room in Finland while the participant had his own room in India.

The test consisted of going through RISE test environment's current features and discussing them. The main focus was to observe the correctness and logics of action sequences. A warm-up task was presented as an easy approach to RISE. Task 1 was again basically the rest of the test.

Warm-up task: Locate Traffica release Z5.4 SP1 and search the list of RTT reports it has available for SGSN adaptation. Start condition: Logged in to RISE on Traffica Main Page. End condition: User has before him NE Adaptation View of SGSN.
Task 1: Add a new field to SGSN's NE release SG8 RTT report. Start condition: User has before him NE Adaptation View of SGSN. End condition: New field has been added to current RTT report.

Table 14. Test tasks.

No additional adaptation data or interface specification was provided, since participant had knowledge over SGSN adaptation which was used as input.

5.5.2. The test results

After some technical difficulties were solved at the beginning of the test, the rest of it went on for most parts fluently and lasted around two hours and fifteen minutes. The scheduled test duration was ninety minutes plus another thirty minutes for feedback. The main reason these time limits were neglected was the fluent interaction that was reached. Still, two hours should be the maximum for this kind of testing, which is how long this test lasted (the first fifteen minutes suffered from technical problems). Encountered technical problems included:

- Failure to connect to internet delayed the start for 10-15 minutes
- Different keyboard/keymap caused user problems typing the input over remote control
- Occasional lag in the web connection showed in page scrolling and updates

The total number of findings reported from the test was 89 (Table 15). Out of these, 21 were usability problems, seven were categorized as other issues, 49 were content-related issues and for the first time bugs were counted too: there were twelve bugs discovered.

Total	Usability problems	Other issues	Content-related issues	Bugs
89	21	7	49	12

Table 15. The number and division of findings.

The number of reported usability problems was 21. Again some of the problems appeared on several views, but were reported only once. There were no critical problems, four serious, six medium and ten low-level usability problems plus one content-related issue.

Severity	Critical	Serious	Medium	Low	Technical / Content
Usability	-	4	6	10	1
Also content-related	-	1	1	4	n/a

Table 16. Usability problems by severity.

The common factor with serious and medium –level problems was that some of the ways the features were presented in GUI were confusing or unnecessarily complicated the completion of common tasks. Others were single mishaps of missing online helps and unclear links. Low-level problems were mostly appearance-related cosmetic issues, but some of them were strongly in connection with the content: mainly the way how certain things were presented and at what context.

The seven other issues were mostly comments on single features of the implementation. These included improving of property names, ideas about version control, error messages and even window scaling issues.

The 49 content-related issues were findings and observations of things that could be done differently. Some comments were improvement ideas some were more speculative on how and why some things could be implemented differently. All these comments were based on user's feedback during the testing.

For this testing the R4T implementation had progressed hugely from previous tests, and for many parts it could be used as the final product would. And while attempting just that, the bugs were discovered. These were features that would not work correctly or were originally specified incorrectly in requirements specification.

5.5.3. Discussion about the results

The fifth usability test for RISE for Traffica was the most challenging so far. The system itself was close to completion of its first version's release and the number of functionalities larger than before, there was need to discover how the whole work process would manage on the system, technical difficulties of the remote connection and also for the first time cross-cultural interaction – all the previous participants had been Finnish.

All the challenges were met however and the test proved to be a success. After the start the interaction worked perfectly, which was helped by the fact that some of the test group were previously familiar with the participant. The amount of feedback surpassed all the previous tests combined and even though the test even went on overtime, it could have still went on. Still, as mentioned before, the two hours seemed as the maximum efficient time for this kind of testing, as people developed fatigue.

The usability problems and their number were well in line with previous test results. Some of the findings were new, and the most serious problems indicated the aspects that required the most attention, such as actual need for solid online helps.

The biggest gain from the test came in the form of content-related issues. These all came from discussions between the members of the test group and the test participant. In fact most of the time the testing was more of discussion than testing as every view and almost every major functionality was talked about. The participant still worked the whole time towards the completion of the task. These comments were important discoveries as they helped develop the system more in the way the actual users would really want it.

This test also showed that the improvements to the expert walkthrough method paid off. The roles that were introduced worked well and every observer now had their own point of view to focus on and take notes. Instead of later interview or discussion after the test every seemingly troublesome feature could be discussed right away when the participant but also all the observers

could ask questions and receive answers to them. This of course required preparations and training of the test group. The test participant was also interested and motivated in helping in the effort the development team was trying to do.

5.6. Personas and scenarios

Creating personas is usually one of the first steps to take in user centered design to be used as the basis for the following work towards scenarios. The current project was well underway until actual persona creation process was started. The need for them was acknowledged from very early on and even the first use cases were produced in the very beginning of the project. This meant that the primal user groups were clear.

Due to the very technical and complex data storage nature of the project, it was however deemed that it would require some implementation before there was an actual need for personas. Therefore the beginning of the project was largely about technical requirements elicitation. The first iterations of implementation aimed at creating the basic framework for the system. The process of creating personas proceeded after RISE for Traffica had evolved to the level where it could be operated through basic GUI.

5.6.1. User groups

The project was started in the first place to improve the tools used by NSN's Traffica developers working on adaptations and thus enhance their working. The goal was to remove unnecessary, time consuming manual labor and testing and allocate the saved time to improve overall quality of adaptations. Thus the primary user group was obvious: Traffica developers.

Creating adaptations for Traffica requires the developers to work in a close collaboration with people developing network elements. Each NE has its own developers and all of these groups might employ different working tools to create interface specifications for NEs. These specifications are typically conveyed to Traffica developers in Excel format.

As the project progressed it became obvious that including network element developers could be mutually beneficial: Traffica developers and network element developers could have one shared tool, and if they both were involved in RISE for Traffica Development that tool would fit both their needs. Therefore the NE developers were selected as the second user group.

The third user group was not quite as obvious as the first two. It was selection between RISE administrators, viewers and customer documentation specialists, or technical writers. The number of technical writers is small, but their work one of the most important in customer interaction and that is why the tertiary user group was decided to be the technical writers.

It was decided to create one persona to present each of these three groups.

5.6.2. Creating personas

For each persona the creation process followed the same steps. The first phase was to collect statistical data of the user group. This data consisted of typical background info such as education, career, competences and problem areas at work. This data was then organized and grouped by the mentioned categories and the essential information was filtered. The second phase was to interview the actual people from the user group to gain more insights into their goals and motivations and to verify the conclusions drawn from the statistical data. After this data was analyzed and compared with the first phase data, a sketch was created for the persona. This sketch was then iterated to add more layers to it and to invent "the untold", persona's personal life. At this time the persona would start to appear almost as a "real person".

The personas were subjected to two evaluation rounds. First it was evaluated and commented by the people the persona was based on. This was to evaluate if something about them seemed odd, or too much out of the ordinary. This gave the personas some details, and served to make them seem more real. The second evaluation was conducted by the NSN's user experience team who had first-hand experience on creating personas to be used in company's other projects. This evaluation provided valuable refinements to personas and crafted them more approachable to software developers.

The idea in the process of creating personas was to be iterative, so that the personas could be refined further if deemed necessary during R4T development project. The personas can be found in the appendices.

The persona for Technical writer was not actual until the first version of RISE for Traffica was released and the work for specifying how to save customer documentation information into database began for the release two. That phase was not reached during the scope of this study and therefore the third persona was not created and is only mentioned here as a future development phase.

5.6.3. Scenarios

Few scenarios were planned for personas during the work towards RISE for Traffica's first release. As the personas were put on hold so were the scenarios as they were intended to model the interaction. Some of the scenarios that were in works included:

For Traffica Developer

- Study of new NE release changes
- Review new NE interface changes
- Find out and update information for new version of NE adaptation
- Export data for interface implementation testing
- Troubleshooting problems and fixing them

For Technical writer

- Review documentation properties
- Checking and fixing documentation properties
- Export customer documentation

An example of scenarios:

Persona: Traffica developer Juhani, involves NE developer Tapani

Scenario: Juhani studies the changes of new NE release

Incidence: Couple of times a year

Lasts: From few days to two weeks

Juhani has some information on what kind of statistics can be measured from a certain Network Element. In case he does not know some specifics, he usually at least has a good idea of whom to ask from NE end. This information comes in handy couple of times a year when new NE release is in works and Juhani has to negotiate with Tapani about the suggested new features for the element.

Revision gives Juhani some idea what the changes are about and he is prepared when some time later he receives the new interface specification from Tapani in his email. Juhani usually reserves a week to study the changes. Sometimes he manages the study faster, sometimes it can take up to two weeks.

He goes through the interface specification documentation comprehensively, every RTT report on field level, to discover all the changes. This part of his work is largely manual labor as he goes through the document, compares it to the previous version, documents the changes (usually to Excel worksheet) and at times discusses with Tapani or his own team members on how to interpret certain changes to specification.

Juhani's study on changes works towards him forming an understanding of the new interface. The document he writes about the changes is both the guideline for the implementation and his own work plan.

Problems: Juhani has problems with changes that occur in the middle of the reports (change of data type), because they cause snowball effect of changes.

5.6.4. Discussion about personas and scenarios

While the work on personas and related scenarios was put on hold very shortly after the work had even begun, the results seemed promising. Two out of three personas were created and approved for use. They were also introduced to the programmers briefly, but in hopes that they would convey the information about who the end users were and how they would use the system in development.

In fact the subcontractor team in Poland was the main target of the personas since they do not have any actual connection with the user groups personas present. With the local Traffica team, the Traffica developers are working in the same premises and thus very familiar. The role of personas was to be more of casting issue: to clarify who actually does what with the system that is to be the basis for scenarios.

The using of personas and thus the scenarios may have been put on hold, but they were not forgotten. They have been mainly used to define the user roles for the system. Besides that, some very simple and short scenarios have been used since and they have employed the personas. While this has been the very minimal usage, it has helped to keep in mind who the designing work is for and that there are people who might think differently from the developing team.

6. Discussion

The work done for this study was two-folded: first to develop RISE for Traffica and second to explore usability methods and develop an approach suitable for implementing usability into software product development. These two topics are first discussed and then this chapter ends with a proposition for usability implementation guideline.

6.1. Discussion about RISE for Traffica development case

During the development cycle for the first release of RISE for Traffica the continuous usability testing was, apart for some occasional try outs of single newly implemented features, the only official testing done for the system by the requirements elicitation team, "the client", until the acceptance tests at the end of the cycle, when R4T 's first version was released. This gave the tests a dualistic nature as the testers needed to report possible technical and logical problems along with usability problems. And more than just report them and come up with solutions to usability problems, the underlying motivation was to gather feedback and ideas for further development of the system.

The first two usability tests conducted on R4T were typical, formative usability tests where participants were briefed about the system and then left to complete the tasks on their own. This method of usability testing provided decent results as the total number of findings from the tests was almost eighty. Almost half of the findings were usability problems. More remarkable, however was the large number of content related and other findings. While some of these were just technical problems, most of them were direct comments and ideas on how to improve the content and the working process with the system.

Employing the lightweight usability methods the next most important step after identifying the problems was to turn them into repair suggestions. This was done by turning issues into action points which each named single action to be completed. The idea was that these action points could be directly added to development board as tasks and assigned to right person. The action points system was introduced in the first usability test and used as such until the third test. The results showed that named action points system worked well, as at least all high and medium level fixes were done by the next test. The table 17 presents how the action points were divided in the first three usability tests.

Test	Type / priority	High	Medium	Low	Total
1st & 2nd	Fix	7	8	10	**25**
	New	2	6	5	**13**
	Study	-	6	5	**11**
3rd	Fix	3	1	2	**6**
	New	-	1	-	**1**
	Study	-	1	4	**5**
Total		**12**	**23**	**26**	**61**

Table 17. The numbers and division of action points from the first three usability tests.

The third usability test was more of an attempt to gain more user insights into R4T development than to discover large number of usability problems. Because of this, the number of findings reduced more than expected. This also meant that the number of action points was lower than before, even though it had been hoped to increase with the number of qualitative feedback. There were few technical problems with the test execution, but not enough to explain the results. It was obvious the changes made to the test arrangements needed to be re-evaluated.

After the third usability test the system experienced the biggest changes so far with the implementation of number of new features and new views. With the possibility for consultation from another experienced usability expert it was decided that best way to analyze new design would be heuristic evaluation. The results discovered some, mostly low level usability problems, but also raised questions about the handling of the content. The timing of heuristic evaluation was fortunate enough for it to act as a complementary to the previous test just as new designs were under way. The results helped to remove some of the problems and clarify few design dilemmas and were considered a success.

By the fifth usability test the system was only a few features short of requirements for the release. Therefore it could be tested more thoroughly than before. The test was properly prepared for and in the end it exceeded all expectations. The number of findings was greater than before and the amount of qualitative feedback in the forms of improvement ideas, comments and questions overwhelming. In this test the participant finally got to experience the whole process of using RISE for Traffica in adaptation creation process and try to understand the new way of working. The most positive feedback was that the process was logical and understandable, and the participant even considered it an improvement. Based on this, the gained feedback was very valuable towards the final tweaking of the system before the release.

The usability tests conducted on RISE for Traffica resulted in total of 226 findings (Table 18). About forty per cent of these were usability problems. Content related issues had about the same share and the rest were categorized under other issues.

Test	Nr of problems	Usability problems	Content related	Others
1st and 2nd	79	37	24	18
3rd	28	14	3	11
Heuristic evaluation	30	19	11	-
5th	89	21	49	19 (incl. bugs)
Total	**226**	**91**	**87**	**48**

Table 18. The number of findings during the usability testing for RISE for Traffica release one.

The purpose of the testing was to discover, identify and come up with solutions to problems with RISE for Traffica system. The system was tested five times, and each test revealed a number of problems. These problems were then analyzed, explained and provided with repair suggestions for the programmers to work on. At first some of the problems repeatedly resurfaced but on the whole they were more often than not repaired. Some repairs were later refined, but they were not listed again as problems unless they were identified as such by test participants.

Table 19 presents the severity of discovered usability problems through the tests. Only the first tests had problems that actually prevented the use of the system. Number of serious usability problems remained constant while the number of medium level issues decreased. The number of low level usability problems increased, which was the result from the development work: the biggest problems were picked out and the new problems were just minor fixes to single feature implementations.

Test / problem severity	Critical	Serious	Medium	Low	Technical/ content
1st and 2nd	3	4	11	7	12
3rd	-	2	3	7	2
Heuristic evaluation	-	1	5	13	11
5th	-	4	6	10	1
Total	**3**	**11**	**25**	**37**	**26**

Table 19. The division of usability problems by severity and test.

After the start the same problems did not reappear again. This means that feedback from the test was efficient and reached the programmers. Table 18 showed that the number of findings was still noticeable even though problems were fixed. This was due the progress of the development: every time new features were implemented, they did not come out flawless but needed some reiteration. Another explanatory fact is that the more the system progressed the higher the number of content related issues became. This is also caused by the developing of the system, as more features were implemented more feedback was received about the whole process of workflow. Problem severity figures from table 19 support the same conclusion.

While the usability work done by testing proved useful for R4T development the work done with personas remained to show its benefits. Two out of three planned personas were created during the project but they were not successfully employed yet. The target was to create scenarios to help with the design of the different views and their functionalities.

RISE is basically a system to store data in. That means many of its functions are tied to certain structures that are constant despite the users. Therefore it was acceptable to postpone the persona creation until the basics of the system were up and running enough to be tested. It can be argued, should the process have started earlier. It is possible because once the system started coming together the development process was so fast, that it soon started requiring design decisions based on user actions, scenarios.

Though not fully employed, the personas still served some purpose as they conveyed information about the end users to the programming team in Poland who had never met them. Personas still remain valid, and can be a good starting point for designs for RISE for Traffica’s next releases.

6.2. Discussion about method development

As discussed above the testing of RISE for Traffica aimed for more than just finding its usability problems: the second goal was to discover any problems related to logic of working process and handling of the adaptation creation's content. This set some requirements for the testing methods too.

At first, however, it was decided to start the testing with assessment of the system. Two rounds of testing with this more traditional approach were conducted. With the second test it became obvious that the used format for usability testing was not enough: the most valuable feedback received from the tests was qualitative as the participants got excited about the product and wanted to discuss it more than just thinking aloud. Analysis of the test results supported the conclusion that in the future the tests need to focus more on conversations than on observations and each side must be able to ask questions and have them answered. In hindsight the need for talking more was obvious due to the new and different nature of working.

The most successful feature from the first tests was the reporting of the problems. Most of the developing team had little or no experience from working with usability issues and the subcontractors in Poland had none. Therefore it was deemed that the best way to convey repair suggestions, i.e. what needs to be done about the findings, is to present them in action points that are easy to add to task boards. Doing this received positive feedback.

For the third usability test the testing method was developed towards what Rubin and Chisnell [2008] call exploratory or formative study. This approach is meant to be used early in the development when there is still a lot of defining and designing happening. The idea was to have the testing process very informal, almost as collaboration between the participant and the moderator. An expert walkthrough approach was then introduced.

The expert walkthrough method was an interesting experiment, but it was not considered a success. By no means was it a failure, but it just did not work quite as well as expected. The major improvements to previous tests were the increased discussion and direct ideas from the participants on how else things could be done. It was decided that the method needed further re-evaluation and redefining.

The third usability test was also the last one with listed action points. This feature had been efficient so far, but by now the team had gained some insight into what usability was about, why these tests were conducted and how to read usability problem descriptions. The separate list of action points was no longer needed.

Work to further improve the expert walkthrough method continued with the fifth usability test. This test was conducted remotely between Finland and India which required more preparations than usual. More than just technical solutions the test crew was also prepared and trained in new way: observers were given distinct roles and focus points. The preparation details went to the level where everyone, especially the technical lead, were required to get to know the system as best they could beforehand and also write down questions about details that might or might not come up.

Despite some technical difficulties during the test, the tweaking of expert walkthrough method paid off as the test was the most successful of them all. In the end the test was not typical usability situation at all, but more of a two-hour discussion about the system. There was a risk of test turning into a chat and not providing any usable results. Despite the length of the session that risk was avoided: the roles introduced for this test worked well, everyone managed to keep their focus on appointed issues and even the participant, who was from the start very positive about the project, managed to carry on the tasks despite getting excited about the possibilities of the system. The analyzed findings proved that the test had been a success with close to ninety different reported issues and improvement ideas.

The expert walkthrough method was developed in order to get more out of a usability test. By the time the need for it arose it had become obvious that summative approaches to testing RISE for Traffica were not enough, nor was there useful quantitative feedback to be gained. So, based on Rubin and Chisnell's [2008] ideas and practices from both cognitive walkthroughs and contextual task analysis the new approach was composed. The first attempt proved that the method has potential, but something more was required. Proper preparations and test task analysis helped some, but in this case the pivotal step was to plan specific roles for the test crew and train everyone to their roles. The test crew had been the same throughout the tests - except for the heuristic evaluation – and despite their inexperience in usability testing, had learned about the goals, methods and process of usability work during the project. With only one usability expert in the team this was the key factor in the success of both the making RISE for Traffica a product of good usability and developing the expert walkthrough a method that could be employed in other projects too.

6.3. Proposition for usability implementation guideline

This work called for practical approach to usability testing that would provide results that are ready to be implemented as a part of the product under development. Gathered here is the summary of the discussion about the key points above that could be used as a framework or a guideline in future endeavors in software development projects:

- Get everyone involved: increasing the usability awareness through training and hands-on experience in the whole team and also among the subcontractors can and will improve the quality of the work.
- Dedicated usability expert in a team helps. Team members should have specific roles and responsibilities and one of these roles should focus on usability.
- Testing with one participant is enough: the nature and quality of the feedback are the most important things. Sometimes an evaluation even without a user can provide the needed feedback providing personas and the most important scenarios have been created.

- Discuss with the participants and listen to their views and suggestions, take notes. Participants should arrive to testing prepared and in optimal cases bring along examples of possible problem cases from their work.
- It is okay to ask. Participants should be encouraged to ask and question the solutions.
- Reserve the necessary time to analyze all the comments and findings, discuss them with the whole team and come up with repair or improvement suggestion for each problem and discovery. The important thing is that all the findings are confirmed and understood.
- Be ready to accept different new ideas and approaches, and adapt them into use. That is the very basis of agile methods also in user experience design: iterations and feedback from users.

7. Conclusion

This study explored how to integrate usability into the development process of RISE for Traffica. R4T is a system that is planned to work as a tool for simplifying and streamlining the adaptation creation process for Traffica. In conjunction with Adaptation Toolkit it offers a completely new way of working with this complex data. The old way of working has been in use for a very long time and with its ups and downs it is what the adaptation specialists have become accustomed to. The challenge discussed in this study has been to develop the new system that answers the users' needs and achieves their acceptance. The key to this has been usability.

The beginning of the usability work on the subject was in the form of summative usability tests. These, while helping to fix the problematic issues of the system, did not provide enough actual information to guide the design. The logical step was to transform the testing method towards more exploratory approach. The first experiment with newly developed expert walkthrough was not a complete success, but the direction was right. After some improvements the new method showed its strengths and the last usability test conducted on R4T's first release version proved to be the most successful of them all.

The case of designing usability for RISE for Traffica worked as a framework for the method development. Developing the expert walkthrough method was one step towards this goal. The ultimate goal was to develop a method or set of methods that could be used in agile software development environment. The context was Nokia Siemens Networks' Traffica program's way of working. The underlying motivation was to come up with simple, fast and inexpensive methods to employ in a product development. Another motivation was to increase the usability awareness within the program during this project.

The usability awareness was never measured, but as discussed earlier, its increase was one the key issues in making the expert walkthrough method a success. The increase in usability awareness was also evident in the Polish programming team, as they quickly adapted to what was new form of feedback from their work.

The studied items show that involving everyone, no matter from what background, from the developing team into usability work has an effect and can alter the ways of working. Even one usability expert in a team can make this happen, but the involvement must be motivational and work towards common goals to achieve the best effect. This of course means that the usability expert must also be active and interested in other team members' task contents.

The most concrete result from the described work was of course RISE for Traffica system. The first release of the system has been delivered and the second version is under way. Its future development engages more the personas that were created but not properly used yet. The continuing work on R4T also follows the guidelines that were outlined as a part of this process. The future of implementing these guidelines to be a part of the Traffica program's way of working is unclear but in the works.

Sources

[Agile Manifesto, 2001] Manifesto for Agile Software Development, www page: http://agilemanifesto.org/, 2001, checked Oct 23rd 2012.

[Chang et al. 2008] Chang, Yen-ning, Lim, Youn-kyong, Stolterman, Erik, Personas: From Theory to Practices, NordiCHI 2008: Using Bridges, 2008.

[Cognitive walkthroughs] Cognitive Walkthroughs, www-page:

http://www.usabilityfirst.com/usability-methods/cognitive-walkthroughs/ , Usability First, checked Feb 21st 2012.

[Connell, 2000] Connell, I.W., Full Principles Set. Set of 30 usability evaluation principles compiled by the author from the HCI literature. Online document at: http://www0.cs.ucl.ac.uk/staff/i.connell/DocsPDF/PrinciplesSet.pdf, checked 30.7.2012.

[Constantine, 2002] Constantine, Larry L., Process Agility and Software usability: Toward Lightweight Usage-Centered Design, reprint from earlier version from Information Age, August/September, 2002.

[Cooper, 1999] Cooper, Alan, The Inmates Are Running the Asylum, Sams, 288 pages, 1999.

[Cooper et al. 2007] Cooper, Alan, Reimann, Robert, Cronin, David, About Face 3, the Essentials of interaction Design, Wiley Publishing Inc., 610 pages, 2007.

[Cunningham, 1992] Cunningham, Ward, Experience Report – The WyCash Portfolio Management System, addendum to the Proceedings of OOPSLA '92, 1992.

[Cunningham, 2011] Cunningham, Ward, Technical Debt, wiki-page, http://c2.com/cgi/wiki?TechnicalDebt, checked Apr 9th 2012.

[Dubberly, 2001] Dubberly, Hugh, Alan Cooper and the Goal Directed Design Process, originally published in GAIN AIGA Journal of Design for the Network Economy Volume 1, Number 2, 2001.

[Expert walkthroughs] Group-based expert walkthrough, www-page: http://www.allaboutux.org/group-based-expert-walkthrough, All about UX, checked Feb 22nd 2012.

[Gerhardt-Powals, 1996] Gerhardt-Powals, Jill, Cognitive engineering principles for enhancing human-computer performance, International Journal of Human-Computer Interaction 8 (2): 189–211, 1996.

[Gould and Lewis, 1985] Gould, John D. and Lewis, Clayton, Designing for Usability: Key Principles and What Designers Think, Communications of the ACM, Volume 28 Issue 3, 1985.

[ISO 92411-11, 1998] Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on Usability, ISO, 1998.

[Kurvinen et al., 2006] Kurvinen, Esko, Aftelak, Andy, Häyrynen, Annakaisa, User-centered Design in the Context of Large and Distributed Projects, CHI EA '06 proceeding, CHI '06 extended abstracts on Human factors in computing system, 2006.

[McInerney & Maurer, 2005] McInerney, Paul and Maurer, Frank, UCD in Agile Projects: Dream Team or Odd Couple? Interactions Magazine Volume 12 Issue 6, 2005.

[Maguire, 2001] Maguire, Martin, Methods to Supports Human-Centred [SIC] Design, International Journal of Human-Computer Studies, volume 55 issue 4, October 2001.

[Medlock et al., 2002] Medlock, Michael C., Wixon, Dennis, Terrano, Mark, Romero, Ramon L., Fulton, Bill, Using the RITE method to improve products; a definition and a case study, Presented at the Usability Professionals Association 2002, Orlando Florida.

[NE, Nokia term bank] Nokia term bank, Network Element, ID4523 Importance: 1 Created: 1991-04-17 Updated: 2007-03-07 Class name: Network management Class: 15 Unit: Networks. Checked 10.2012.

[Nielsen, 1989] Nielsen, Jakob, Usability Engineering at a Discount, Proceedings of the third international conference on human-computer interaction on Designing and using human-computer interfaces and knowledge based system, 1989.

[Nielsen and Molich, 1990] Nielsen, Jakob and Molich, Rolf, Heuristic Evaluation of User Interfaces, CHI '90 Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people, 1990.

[Nielsen, 1994a] Nielsen, Jakob, Usability Engineering, new edition, Morgan Kaufmann, 362 pages, 1994.

[Nielsen, 1994b] Nielsen, Jakob, Guerrilla HCI: Using Discount Usability Engineering to Penetrate the Intimidation Barrier, www-article (http://www.useit.com/papers/guerrilla_hci.html), 1994. Checked 30.11.2011

[Nielsen, 1994c] Nielsen, Jakob, Ten Usability Heuristics a list, www-page: http://www.useit.com/papers/heuristic/heuristic_list.html, 1994, checked Mar 5th 2012.

[Nielsen, 2009] Nielsen, Jakob, Discount usability: 20 Years, Jakob Nielsen's Alertbox, September 14, 2009, www-page: http://www.useit.com/alertbox/discount-usability.html, checked 30.7.2012.

[Nodder and Nielsen, 2008] Nodder, Chris and Nielsen, Jakob, Agile Usability: Best Practices for User Experience on Agile Development Projects, 1st edition, Nielsen Norman Group, 95 pages, 2008.

[Pihlajamäki, 2010] Pihlajamäki, Mikko, Improving usability in agile software development process of network monitoring system, Master of Science Thesis, Tampere University of Technology, 88 pages, April 2010.

[Pruitt and Grudin, 2003] Pruitt, John and Grudin, Jonathan, Personas: practice and Theory, DUX ’03 Proceedings of the 2003 conference on Designing for user experience, 2003.

[RISE Documentation] NSN Document Number: D-166113.

[Rubin and Chisnell, 2008] Rubin, Jeffrey and Chisnell, Dana, Handbook of Usability Testing – How to Plan, Design, and Conduct Effective Tests, second edition, Wiley Publishing Inc., 348 pages, 2008.

[Silva da Silva et al. 2011] Silva da Silva, Tiago, Martin, Angela, Maurer, Frank, Silveira, Milene, User-Centered Design and Agile Methods: A Systematic Review, Proceedings of the International Conference on Agile Methods in Software Development (Agile 2011), Salt Lake City, UT, 2011.

[Sohaib and Khan, 2011] Sohaib, Osama and Khan, Khalid, Incorporating Discount Usability in Extreme Programming, International Journal of Software Engineering and Its Applications Vol. 5 No 1, January, 2011.

[TRUMP] TRUMP, www-page: http://www.usabilitynet.org/trump/, checked 19.7.2012.

[Wolkerstorfer et al., 2008] Wolkerstorfer, Peter, Tscheligi, Manfred, Sefelin, Reinhard, Milchrahm, Harald, Hussain, Zahid, Lechner, Martin, Shahzad, Sara, Probing an Agile Usability Process, proceedings of CHI EA ’08 Extended Abstracts on Human Factors in Computing Systems, 2008.

Appendices

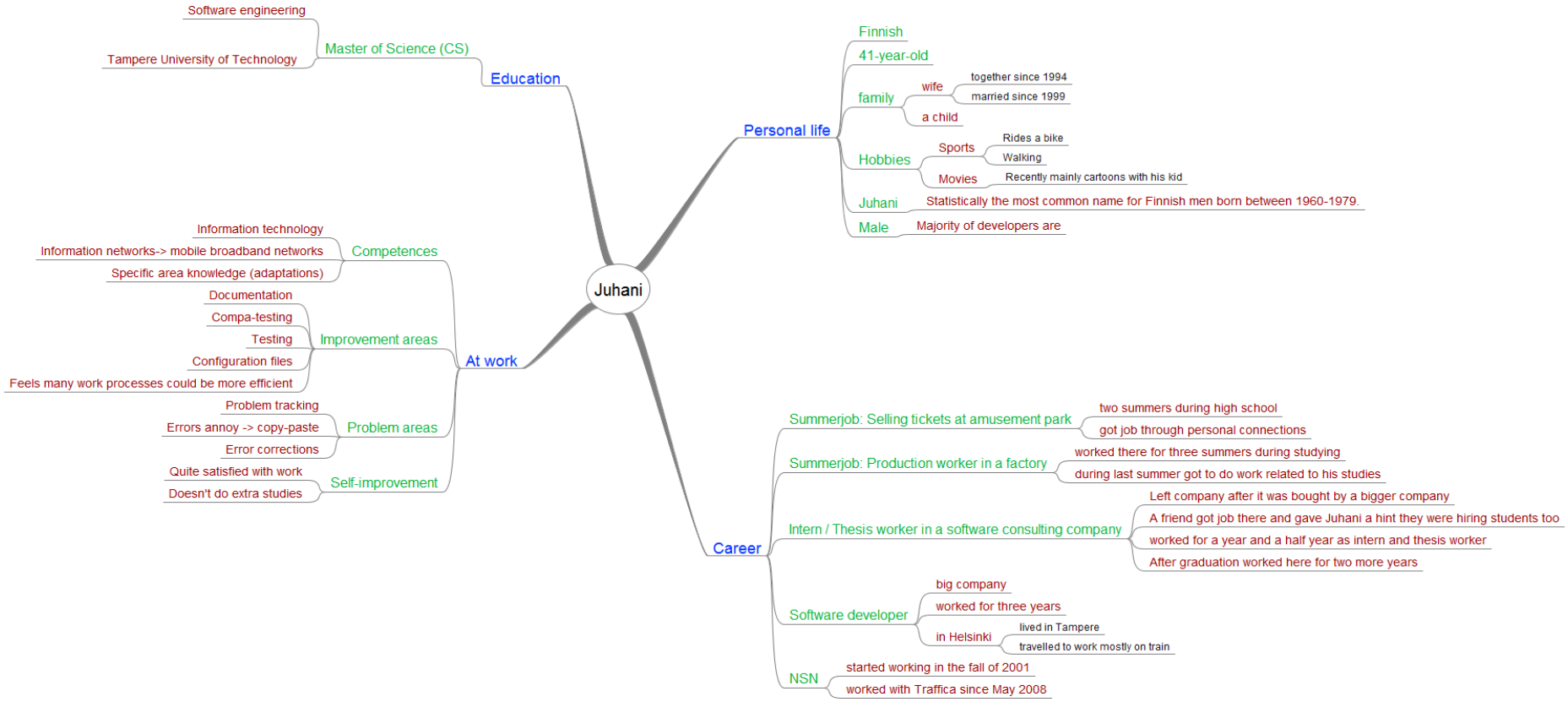
Initial mind map for Traffica Developer

Primary persona Traffica Developer

Initial mind map for Network Element Developer

Secondary persona Network Element Developer

Appendix 1: Initial Mind map for Traffica Developer



Appendix 2: Primary persona Traffica Developer

1.1 Juhani Virtanen, 41, Traffica developer

"If I had some prediction of coming changes I could optimize my work."

Juhani's background

Juhani is a 41-year-old senior specialist working with Traffica for Nokia Siemens Networks. He worked for other ITC companies before but he's been working for NSN for more than 10 years and is quite at home in the company. He has worked on Traffica since May 2008.

During his time with Traffica Juhani has become an adaptation specialist. The majority of other specialists work in the same office with him, so Juhani knows them all very well. And while he specializes in single adaptation, he also has good understanding what the different adaptations are about.

Juhani's main work is to push and react to changes in the latest interface specification provided by Network Element developers who are work on different sites. Conveying all the information requires a lot of interaction so Juhani spends plenty of time handling email and calls. He studies how planned changes effect to interface to avoid unwanted impacts.

After NE developer has delivered interface specification, begins Juhani's busiest work. In a common case he has between 4 to 6 weeks to study what changes have been made and how they effect to configuration files.

During this time Juhani implements the changes to RTT reports. This means a lot of work since he has to search for and change the information from and to various sources and locations. This is a time consuming process as there can be tens of reports that need updating. Juhani doesn't really mind these manual tasks, but his manager feels that this work step could be optimized.

Besides implementing the changes Juhani tests them for possible unwanted impacts on Traffica functionality and compatibility. Due to recent change to Kanban method Juhani's time spent on testing is reduced, but he feels the testing phase could still be improved, e.g. with automatically generated test data.

The most annoying thing about Juhani's work is the annoying tracking of errors as simple as typing errors. Sometimes he feels he has inadequate knowledge on some details, which can frustrate him.

Juhani's job is stressful but he's learned to let off the steam by cycling home - he finds that meditating.

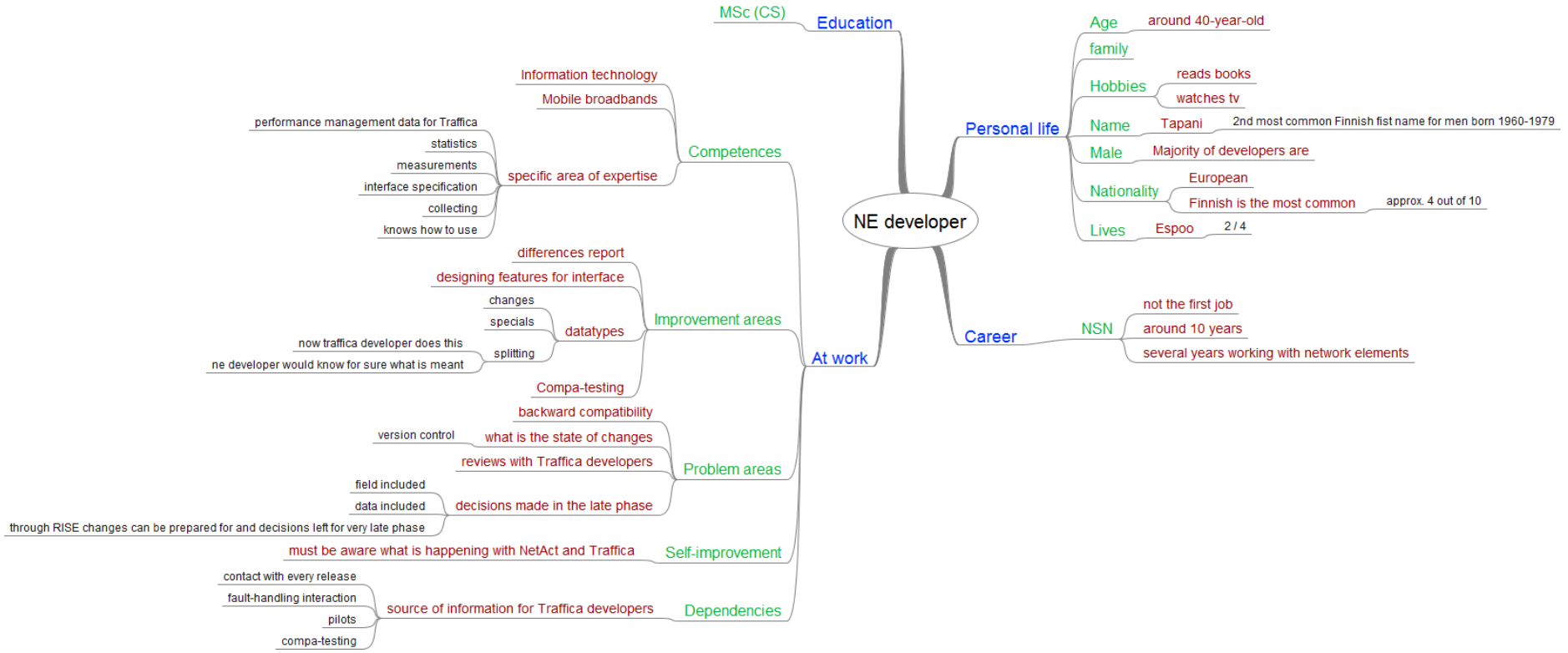
Juhani's goals

1. Find all adaptation information in one place. This would save time for more important tasks. Currently required information needs to be searched from and updated to several and dispersed files.

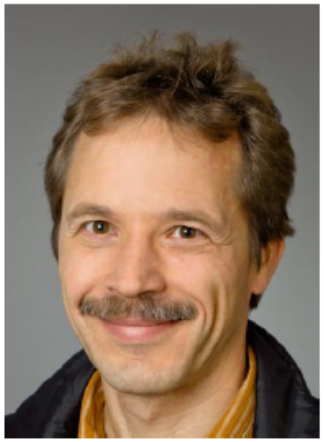
2. Spend less time on testing. Optimized testing would save time for resolving issues with interface specifications and better descriptions.

3. Communicate with concerning parties. Juhani needs to be informed of coming changes by Network Element developer. His work, in turn, is the basis for customer documentation.

Appendix 3: Initial Mind map for Network Element Developer



1.2 Tapani Korhonen, 42, Network Element developer



"Wait a minute while I check the details from email."

Tapani's background

Tapani is a 42-year-old Network Element developer. He has been working for NSN for more than ten years focusing on mobile broadbands. He has several years of experience on working with network elements and he is one of around ten persons with profound knowledge on this specific area.

Tapani is usually quite busy with his work. He must be aware of what is happening with Traffica, but is only a part of his work as he has responsibilities with NetAct interface too. The main thing in Tapani's work that affects Traffica development is designing and updating interface specifications in collaboration with Traffica developers. His other Traffica related tasks include specifying and doing measurements and collecting statistics for performance management data.

Tapani's cue to start creating new interface is the new features for Network Element. He starts discussions concerning visibility and inclusion of new fields with Traffica developers. These discussions mainly take place over email. The process of refining the interface specification lasts throughout development increments until the specification is frozen.

During discussion many email attachments change hands since NE developers and Traffica don't always share uniform documentation. There may also be occasions when Traffica developers work ahead and Tapani's work, that should be guideline for them, lags behind.

Tapani feels reviews with Traffica developers are very useful. He assumes some of his work could be simpler if they shared a common tool. Tapani wishes he could do some of this work rather earlier than later phase of development. He hopes for tools for designing features for interface in earlier phase of production and following the states of proposed changes. These might relieve the pressure from the late decisions of what to include that are so typical for Traffica releases.

Tapani's goals

1. Have a way to report, compare, review and communicate changes between releases. RISE 4 Traffica may provide single tool for the job if it is accepted for use. Common tool may also simplify communication.

2. Have awareness of and control over states of changes. Tapani wants to know when to freeze and what is the agreed frozen interface version. He also informs Traffica developers when the interface is frozen.

3. Have flexibility on creating new NE release interface specifications. Tapani would like to use the same tool for designing which could mean more flexibility on time tables as well.